

EVALITA 2026

**The Ninth Evaluation Campaign of Natural Language
Processing and Speech Tools for Italian**

Proceedings of the Workshop

February 26-27, 2026

Copyright ©2026 for the individual papers by the papers' authors.
Copyright ©2026 for the volume as a collection by its editors.
This volume and its papers are published under the Creative Commons License Attribution 4.0 International (CC BY 4.0).

These proceedings have been published in the CEUR Workshop Proceedings series.
The original papers are available at: <https://ceur-ws.org/Vol-4195>.
The papers are mirrored in the ACL Anthology.

ISBN 979-12-243-0586-6

Table of Contents

Preface

Francesco Cutugno, Alessio Miaschi, Alessio Palmero Aprosio, Giulia Rambelli, Lucia Siciliani and Marco Antonio Stranisci	1
---	---

EVALITA 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian

Francesco Cutugno, Alessio Miaschi, Alessio Palmero Aprosio, Giulia Rambelli, Lucia Siciliani and Marco Antonio Stranisci	5
---	---

Tu chiamale se vuoi emozioni: The impact of emotions on disinformation and sexism identification

Paolo Rosso	14
-----------------------	----

Kenji-Endo: a BabyLM @EVALITA

Calogero Jerik Scozzaro, Matteo Rinaldi, Gianluca Mittone and Marco Antonio Stranisci	16
---	----

DeSegMa-IT at EVALITA 2026: Overview of the "Detection and Segmentation of Machine Generated Texts in Italian"

Giovanni Puccetti, Andrea Pedrotti and Andrea Esuli	27
---	----

Gradient Descenders at DeSegMa-IT: Leveraging Monolingual Transformer for LLM-Generated Text Detection and Boundary Identification

Tran Phuoc Thanh Nhan, Bui Hong Son and Dang Van Thin	42
---	----

MINDS at DeSegMa-IT: Detecting Human-LLM Authorship Switches via Token-Level Classification

Flavio Giobergia	53
----------------------------	----

Nicla at DeSegMa-IT: DistilBERT for MGT Detection and LightGBM Regressor for HMT Segmentation

Nicla Auletta	59
-------------------------	----

Stochastic Gradient Descenders at DeSegMa-IT: Instruction-Tuned LLM and Token Classification for MGT Detection and Boundary Localization

Huynh Nguyen Phu, Bui Hong Son and Dang Van Thin	67
--	----

UniTor at DeSegMa-It: Analyzing Supervision and Encoder Representations for Italian Machine-Generated Text Detection

Federico Borazio, Giacomo De Luca, Daniele Pasquini, Danilo Croce and Roberto Basili	78
--	----

GSI:detect at EVALITA 2026: Overview of the Task on Detecting Gender Stereotypes in Italian

Gloria Comandini, Manuela Speranza, Sofia Brenna, Davide Testa, Stefania Cavagnoli and Bernardo Magnini	96
---	----

DIAG-Sapienza at GSI:detect: Joint Detection and Classification of Gender Stereotypes with Structured Prompting and Fine-Tuning

Olga E. Sorokoletova, Emanuele Musumeci and Daniele Nardi	109
---	-----

Festa at GSI:detect: In-Context Learning vs. Fine-Tuning for Gender Stereotype Detection

Donato Festa	123
------------------------	-----

MINDS at GSI:detect: From Logits to Degrees of Agreement in Gender Stereotype Detection with LLMs

Flavio Giobergia	131
----------------------------	-----

<i>StereoBusters at GSI: detect: LLM-Based Detection and Human Qualitative Analysis of Gender Stereotypes in Italian Short Text</i>	
Salvatore Greco, Moreno La Quatra, Marta Marchiori Manerba, Ricardo Muñoz Sánchez and Alessandra Teresa Cignarella	138
<i>Team Prisma at GSI: detect: Comparing PB&J Persona-Based and Few-Shot Approaches to Gender Stereotype Detection at EVALITA 2026</i>	
Claudia Zaghi	157
<i>Tiz at GSI: detect: Modeling Gender Stereotype Detection as Multi-Category Gender Stereotype Scoring</i>	
Tiziano Labruna	169
<i>MultiPRIDE at EVALITA 2026: Overview of the Multilingual Automatic Detection of Slur Reclamation in the LGBTQ+ Context Task</i>	
Chiara Ferrando, Lia Draetta, Marco Madeddu, Mae Sosto, Viviana Patti, Paolo Rosso, Cristina Bosco, Jacinto Mata and Estrella Gualda	178
<i>AIWizards at MULTIPRIDE: A Hierarchical Approach to Slur Reclamation Detection</i>	
Luca Tedeschini and Matteo Fasulo	196
<i>AVAHI at MultiPRIDE: Multilingual Reclaimed Language Detection via Knowledge Graphs and Retrieval-Augmented Generation</i>	
Tania Alcántara, Omar García-Vázquez, José A. Torres-León, Marco Cardoso-Moreno, Diana Jiménez and Luis Moreno-Mendieta	206
<i>Challenger at MultiPRIDE: Is It Hate Speech or Reclaimed?</i>	
Hadi Bayrami Asl Tekanlou, Mahdi Bakhtiyarzadeh and Jafar Razmara	229
<i>DataSummit at MultiPRIDE: Context-Aware Multilingual Detection of Slur Reclamation in LGBTQ+ Contexts</i>	
Federico Dingeo and Marco Viviani	238
<i>Ghavidel-Rajabi at MultiPRIDE: Identity, Toxicity, or Complexity? A Language-Specific Feature Selection Approach to Reclamatory Intent Detection</i>	
Houman Rajabi, Fatemeh Ghavidel and Kourosh Ghahremani	251
<i>GRUPPETTOZZO at MultiPRIDE: Detecting LGBTQ+ Reclamatory Intent via Context-Aware Transformers</i>	
Federico Traina, Alessandro Santoro, Gabriele Greco, Irene Siragusa and Roberto Pirrone	262
<i>HateItOff at MultiPRIDE: Linguistic and Sentiment Cues in Reclaimed LGBTQ+ Slur Detection</i>	
Greta Damo and Nicolás Benjamín Ocampo	273
<i>I2C At MultiPRIDE: Transformers And Generative LLMs For Multilingual Reclamation Classification</i>	
Laura Vázquez-Ramos, Jacinto Mata-Vázquez and Victoria Pachón-Álvarez	284
<i>INFOTEC-NLP at MultiPRIDE: Region-Aware Transformer Models for Spanish Reclaimed Language Classification</i>	
Jorge Gleaves and Guillermo Ruiz	294
<i>KIT-TIP-NLP at MultiPride: Continual Learning with Multilingual Foundation Model</i>	
Barathi Ganesh HB, Michal Ptaszynski, Rene Melendez and Juuso Eronen	302
<i>LlaNa at MultiPRIDE: A Fine-Tuning Approach with Cross-Lingual Augmentation</i>	
Alessio Mercurio, Giorgio Talluto, Irene Siragusa and Roberto Pirrone	315

<i>MilaNLP at MultiPRIDE: Evaluating Lexical, Transformer, and Retrieval-Augmented Models for Reclaimed Language</i>	
Ivana Crescenzi, Arianna Muti and Debora Nozza	322
<i>NamDang at MultiPride@EVALITA 2026: Multilingual Classification of Reclaimed Language in LGBTQ+ Discourse using Transformer-based Models</i>	
Nam Dang, Vo Tuan Kiet and Dang Van Thin	335
<i>NetGuardAI at MultiPRIDE: Multilingual Detection of Reclaimed Language</i>	
Noe Come Jacques Le Pollotec, Elena-Simona Apostol and Ciprian-Octavian Truică	344
<i>SaFe Tweets at MultiPRIDE: From Multi-Model Ensembles to Expert LLMs for Reclamation Detection</i>	
Sara Visconti and Federica Manzi	352
<i>The Hate Busters at MultiPRIDE: Automatic Identification of Reappropriated Slurs in Multilingual LGBTQ+ Discourse</i>	
Aurora Ciminelli, Giulia Corvino, Camilla Gentili and Marco Viviani	366
<i>UniBO-FICLIT at MultiPRIDE: Fine-Tuning an ELECTRA-Based Model for the Detection of Italian Reclaimed Slurs</i>	
Simone Casazza	383
<i>Cruciverb-IT at EVALITA 2026: Overview of the Crossword Solving in Italian Task</i>	
Cristiano Ciaccio, Gabriele Sarti, Alessio Miaschi, Felice Dell’Orletta and Malvina Nissim . .	392
<i>AC/DG at Cruciverb-IT: Retrieval-Based Approaches for Italian Crossword Clue Answering</i>	
Ali Yassine, Claudio Savelli, Davide Napolitano, Giuseppe Gallipoli, Luca Cagliero and Elena Baralis	410
<i>FFT-UniBa at Cruciverb-IT: Special Length Tokens and CSP for Italian Crossword Solving</i>	
Andrea Porcelli, Filippo Di Gravina, Emanuele Fontana, Mattia Curri and Francesco Damiano Di Gregorio	421
<i>MINDS at Cruciverb-IT: Solving Italian Crossword Clues with Masked Language Models and Candidate Pooling</i>	
Flavio Giobergia	433
<i>UNIBA at Cruciverb-IT: Solving Italian Crosswords with Encoder–Decoder Models and Beam Search</i>	
Pierpaolo Basile	442
<i>UniTor at Cruciverb-IT: Retrieval-Augmented Two-Step Reasoning for Italian Crossword Clue Answering</i>	
Andriy Shcherbakov, Danilo Croce and Roberto Basili	452
<i>SVELA at EVALITA 2026: Overview of the Selective Verification of Erasure from LLM Answers Task</i>	
Claudio Savelli, Moreno La Quatra, Alkis Koudounas and Flavio Giobergia	467
<i>Eraserhead at SVELA: Detecting LLM Forgetting via Logit-Space Statistics</i>	
Matteo Berta and Tania Cerquitelli	477
<i>Ita-Lib at SVELA: Detecting the Forgotten — Representation-Based Approach for Verifying Machine Unlearning</i>	
Ali Yassine, Hadi Ibrahim and Luca Cagliero	484
<i>MALTO at SVELA: A Specific-Attention-Head Approach for Membership Inference Attacks in LLMs Unlearning Evaluation</i>	
Evren A. Munis, Mattia Sabato, Erfan Bayat and Andrea Lolli	495

<i>priyam_saha17 at SVELA: A Feature-Centric Pipeline for Verifying Selective Forgetting in Large Language Models</i>	
Priyam Saha	509
<i>EVWSD-ITA at EVALITA 2026: Overview of the Enhanced Visual Word Sense Disambiguation for Italian Task</i>	
Elio Musacchio, Lucia Siciliani, Pierpaolo Basile and Giovanni Semeraro	522
<i>UniTor at EVWSD-ITA: Zero-Shot Visual Word Sense Disambiguation via Visual Question-Answering</i>	
Claudiu D. Hromei, Antonio Scaiella, Danilo Croce and Roberto Basili	532
<i>ATE-IT at EVALITA 2026: Overview of the Automatic Term Extraction Italian Testbed Task</i>	
Nicola Cirillo, Giorgio Maria Di Nunzio and Federica Vezzani	550
<i>DSBKTE at ATE-IT: From Token Classification to Zero-Shot Generation: Two Approaches to Italian ATE at EVALITA 2026</i>	
Danil Smirnov, Bitu Khashechian and Giorgio Maria Di Nunzio	563
<i>JTTE at ATE-IT: A CRF Model with Contextual Embeddings</i>	
Juliette Tonneau and Giorgio Maria Di Nunzio	572
<i>MKTE at ATE-IT: CRF-Based Term Extraction for Italian Waste Management Documents</i>	
Minseok Kim and Giorgio Maria Di Nunzio	581
<i>OATE at ATE-IT: A Hybrid Approach for Automatic Terminology Extraction in Italian</i>	
Omar Arab and Giorgio Maria Di Nunzio	587
<i>Peacemaker at ATE-IT: Automatic term extraction from Italian text for waste management data using encoder model</i>	
Mahdi Bakhtiyarzadeh, Hadi Bayrami Asl Tekanlou and Jafar Razmara	593
<i>SMTE at ATE-IT: Ensemble Term Extraction with Italian BERT, spaCy, and Vocabulary-Based Filtering</i>	
Sofia Maule and Giorgio Maria Di Nunzio	602
<i>TermNinjas at ATE-IT: Overview of the Term Extraction and Term Variants Clustering Task</i>	
Zunaira Hasnain and Fashad Ahmed Siddique	614
<i>TrietNLP at ATE-IT: A Hybrid Pipeline for Italian Waste Management Terminology Analysis</i>	
Nguyen Minh Triet and Dang Van Thin	623
<i>VVTE at ATE-IT: From Candidates to Terms: Hybrid Italian ATE with Dependency Heuristics, Gemini, and Random Forest Filtering</i>	
Valentine G. L. Vandervoort and Giorgio Maria Di Nunzio	633
<i>PFB at EVALITA 2026: Overview of the Prometeia Financial Benchmark</i>	
Alessandro P. Bardelli, Tolga Çekiç, Irem Demirtaş, Michele Filannino, Simona Scala, Andrea Galassi, Gianmarco Pappacoda and Paolo Torroni	639
<i>AMSN at PFB: Overview of the Automated Multilingual Economics Question Answering Task</i>	
Ali Saleh Mohammadabad and Mehregan Nazarmohsenifakori	646
<i>UniTor at PFB: Constrained Agentic Prompting and Self-Consistency for Financial Multi-Choice QA</i>	
Federico Borazio, Seyed Alireza Mousavian Anaraki, Shahid Iqbal Rai, Danilo Croce and Roberto Basili	654
<i>FadeIT at EVALITA 2026: Overview of the Fallacy Detection in Italian Social Media Texts Task</i>	
Alan Ramponi and Sara Tonelli	673

<i>Label at FadeIT: Fallacy-Aware LLM Reasoning for Score-Based Classification</i>	
Tiziano Labruna and Eleni Papadopulos	686
<i>MALTO at FadeIT: A BERT-Based System for Multi-Label Fallacy Detection in Italian Social Media</i>	
Matin Salami, Luca M. Rodia, Vladimir Schiau, Evren A. Munis, Claudio Savelli and Flavio Giobergia	696
<i>PuDy at FadeIT: Enhancing Fine-Grained Fallacy Detection with Hierarchy-Aware Training and Contextual Enrichment</i>	
Duy Dang Phu, Son Bui Hong and Dang Van Thin	709
<i>RBG-AI at FadeIT: Prompted LLMs with Label Abstraction for Logical Fallacy Detection</i>	
Meenakshi, Jairam R, Reshma U, Barathi Ganesh HB and Michal Ptaszynski	721
<i>TIGRO at FadeIT: E Pluribus Unum – A Multi-task Approach to Fallacy Detection and Span Identification</i>	
Stefano Atzeni, Gabriele Sarti, Tommaso Caselli and Malvina Nissim	734
<i>Unica at FadeIT: Adapting Large Language Models to Fallacy Identification in Social Networks</i>	
Matteo Fenu and Maurizio Atzori	748
<i>IMPOLS at EVALITA 2026: Overview of the IMPOLS Task</i>	
Lorenzo Gregori, Walter Paci and Valentina Saccone	759
<i>RES2 at IMPOLS 2026: Implicit Content Classification via Span-Awareness, Hierarchical Multi-Task Learning and Synthetic Data</i>	
Salvatore Spezia, Salvatore Ferrara, Roberto Dioguardi, Ernesto Davì, Irene Siragusa and Roberto Pirrone	770
<i>INDAQ2 - A Large Italian Narrative QA Benchmark: A CALAMITA 2026 Challenge</i>	
Luca Gioffré, Luca Moroni, Alberte Fernández-Castro, Elena Marafatto, Giacomo Garufi and Roberto Navigli	782
<i>ProverbIT - Easy to complete, hard to choose: A CALAMITA Challenge</i>	
Enrico Mensa, Lorenzo Zane, Calogero Jerik Scozzaro, Matteo Delsanto, Tommaso Milani and Daniele P. Radicioni	806

Preface to the EVALITA 2026 Proceedings

Francesco Cutugno¹, Alessio Miaschi², Alessio Palmero Aprosio³, Giulia Rambelli⁴, Lucia Siciliani⁵ and Marco Antonio Stranisci^{6,7}

¹Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy

²Institute for Computational Linguistics "A. Zampolli" (CNR-ILC) - ItaliaNLP Lab, Pisa, Italy

³University of Trento, Trento, Italy

⁴Aix-Marseille University | ILCB, Aix-en-Provence, France

⁵Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

⁶Università degli Studi di Torino, Torino, Italy

⁷aequa-tech, Torino, Italy

1. Preface

EVALITA 2026 is an initiative of AILC (Associazione Italiana di Linguistica Computazionale)¹, and it is endorsed by the Italian Association for Artificial Intelligence (AIxIA)² and the Italian Association for Speech Sciences (AISV)³. As in the previous editions (<http://www.evalita.it/>), EVALITA 2026 is organized along a set of selected tasks, which provide participants with opportunities to discuss and explore both emerging and traditional areas of Natural Language Processing (NLP) and Speech for Italian. The participation is encouraged for teams working both in academic institutions and industrial organizations.

This volume includes the reports of both task organizers and participants to all of the EVALITA 2026 challenges. In the 2026 edition, we coordinated the organization of 10 different tasks belonging to five research areas, being:

Authorship Analysis

- *DeSegMa-It: Detection and SEgmentation of MACHine generated texts in Italian* (G. Puccetti, A. Pedrotti, A. Esuli)

Computational Ethics

- *GSI: detect – Detecting Gender Stereotypes in Italian* (G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini)
- *MultiPRIDE: Multilingual Automatic Detection of Reclamation of Slurs in the LGBTQ+ Context* (C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda)

Emerging Tasks

- *Cruciverb-IT: Crossword Solving* (C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, M. Nissim)
- *SVELA: Selective Verification of Erasure from LLM Answers* (C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia)

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ cutugno@unina.it (F. Cutugno); alessio.miaschi@ilc.cnr.it (A. Miaschi); a.palmeroaprosio@unitn.it (A. P. Aprosio); giulia.RAMBELLI@univ-amu.fr (G. Rambelli); lucia.siciliani@uniba.it (L. Siciliani); marcoantonio.stranisci@unito.it (M. A. Stranisci)

✉ 0000-0001-9457-6243 (F. Cutugno); 0000-0002-0736-5411 (A. Miaschi); 0000-0002-1484-0882 (A. P. Aprosio); 0000-0003-3379-6941 (G. Rambelli); 0000-0002-1438-280X (L. Siciliani); 0000-0001-9337-7250 (M. A. Stranisci)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://www.ai-lc.it/en/>

²<https://aixia.it/en/>

³<https://www.aisv.it/>

Multimodality

- *EVWSD-ITA: Enhanced Visual Word Sense Disambiguation for Italian* (E. Musacchio, L. Siciliani, P. Basile, G. Semeraro)

New Challenges in Long-Standing Tasks

- *ATE-IT: Automatic Term Extraction – Italian Testbed* (N. Cirillo, G. M. Di Nunzio, F. Vezzani)
- *PFB: Prometeia Financial Benchmark – Benchmarking Language Models in the Financial Domain* (A. P. Bardelli, T. Çekiç, I. Demirtaş, M. Filannino, S. Scala, A. Galassi, G. Pappacoda, P. Torroni)

Pragmatics

- *FadeIT: Fallacy Detection in Italian Social Media Texts* (A. Ramponi, S. Tonelli)
- *IMPOLS: IMplicit contents in POLitical Speech* (L. Gregori, W. Paci, V. Saccone)

The workshop took place in Bari on 26–27 February 2026 and provided a valuable opportunity for participants from both academia and industry to present their work, exchange ideas, and discuss results within task-oriented sessions as well as during the final plenary discussion. As in previous editions, we continued the tradition of the Best System award, which this year was dedicated to the memory of Federico Sangati. The award aimed to encourage students, developers, and researchers to explore innovative approaches and to challenge the current state of the art, regardless of the final outcome.

We would like to express our sincere thanks to our supporters, AI2B⁴, Aptus.ai⁵ and Prometeia⁶, and to the University of Bari “Aldo Moro” for hosting the event. We are also grateful to the members of the “Federico Sangati” Best System Across Tasks Awards committee (Dominique Brunato, Camilla Casula, Alessandro Mazzei, Marco Polignano) for their commitment and expertise. Our thanks also go to the AILC Board for their continued trust and support. We warmly thank our invited speaker, Paolo Rosso, for sharing his experience and insights, and we finally acknowledge all task organizers and participants whose enthusiasm and creativity made this edition particularly successful.

Chairs

Francesco Cutugno (University of Naples)

Alessio Miaschi (Istituto di Linguistica Computazionale “A. Zampolli” – CNR)

Alessio Palmero Aproso (University of Trento)

Giulia Rambelli (Aix-Marseille University)

Lucia Siciliani (University of Bari)

Marco Antonio Stranisci (University of Turin)

Steering Committee

Alan Ramponi (Fondazione Bruno Kessler)

Alessandro Maisto (University of Salerno)

Alessio Miaschi (Istituto di Linguistica Computazionale “A. Zampolli” – CNR)

Alessio Palmero Aproso (Università di Trento)

Alkis Koudounas (Politecnico di Torino)

Andrea Esuli (Istituto di Scienza e Tecnologie dell’Informazione “A. Faedo”, CNR-ISTI)

Andrea Pedrotti (Istituto di Scienza e Tecnologie dell’Informazione “A. Faedo”, CNR-ISTI)

Chiara Ferrando (University of Turin)

⁴<https://www.ai2b.it/>

⁵<https://aptus.ai/it/>

⁶<https://www.prometeia.com/it>

Claudio Savelli (Politecnico di Torino)
Cristiano Ciaccio (Istituto di Linguistica Computazionale “A. Zampolli” – CNR)
Cristina Bosco (University of Turin)
Danilo Croce (University of Rome Tor Vergata)
Davide Testa (Fondazione Bruno Kessler)
Elio Musacchio (University of Bari)
Federica Vezzani (University of Padua)
Felice Dell’Orletta (Istituto di Linguistica Computazionale “A. Zampolli” – CNR)
Flavio Giobergia (Politecnico di Torino)
Gabriele Sarti (Northeastern University)
Giorgio Di Nunzio (University of Padua)
Giovanni Puccetti (Istituto di Scienza e Tecnologie dell’Informazione “A. Faedo”, CNR-ISTI)
Giulia Rambelli (Aix-Marseille University)
Gloria Comandini (University of Milan)
Jacinto Mata (Universidad de Huelva)
Lia Draetta (University of Turin)
Lorenzo Gregori (University of Florence)
Lucia Siciliani (University of Bari)
Mae Sosto (Centrum voor Wiskunde en Informatica)
Malvina Nissim (University of Groningen)
Manuela Speranza (Fondazione Bruno Kessler)
Marco Antonio Stranisci (University of Turin)
Marco Madeddu (University of Turin)
Moreno La Quatra (Kore University of Enna)
Nicola Cirillo (University of Salerno)
Pierpaolo Basile (University of Bari)
Sara Tonelli (Fondazione Bruno Kessler)
Simona Scala (Prometeia)
Sofia Brenna (Free University of Bozen-Bolzano)
Valentina Saccone (University of Florence)
Viviana Patti (University of Turin)
Walter Paci (University of Florence)

CALAMITA Chairs

Pierpaolo Basile (University of Bari)
Danilo Croce (University of Rome Tor Vergata)
Malvina Nissim (University of Groningen)
Viviana Patti (University of Turin)

CALAMITA Data and Evaluation Team

Giuseppe Attanasio (Instituto de Telecomunicações, Lisbon), Coordinator
Federico Borazio (University of Rome, Tor Vergata)
Maria Francis (University of Groningen; University of Trento)
Jacopo Gili (University of Turin)
Elio Musacchio (University of Bari Aldo Moro)
Matteo Rinaldi (University of Turin)
Daniel Scalena (University of Groningen; University of Milan Bicocca)

Proceedings Chairs

Alessio Miaschi (Istituto di Linguistica Computazionale “A. Zampolli” – CNR)

Marco Antonio Stranisci (University of Turin)

“Federico Sangati” Award Committee

Dominique Brunato (Istituto di Linguistica Computazionale “A. Zampolli” – CNR)

Camilla Casula (Fondazione Bruno Kessler)

Alessandro Mazzei (University of Turin)

Marco Polignano (Independent Researcher)

Website

Alessio Palmero Aprosio (University of Trento)

Sara Barcena (Freelance designer)

Booklet and badge

Eleonora Ghizzota (University of Bari)

Elio Musacchio (University of Bari)

Lucia Siciliani (University of Bari)

Bari, February 2026

EVALITA 2026: Overview of the 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian

Francesco Cutugno¹, Alessio Miaschi², Alessio Palmero Aprosio³, Giulia Rambelli⁴, Lucia Siciliani⁵ and Marco Antonio Stranisci^{6,7}

¹Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy

²Institute for Computational Linguistics "A. Zampolli" (CNR-ILC) - ItaliaNLP Lab, Pisa, Italy

³University of Trento, Trento, Italy

⁴Aix-Marseille University | ILCB, Aix-en-Provence, France

⁵Department of Computer Science, University of Bari Aldo Moro, Bari, Italy

⁶Università degli Studi di Torino, Torino, Italy

⁷aequa-tech, Torino, Italy

Abstract

EVALITA is a shared evaluation campaign designed to assess and compare Natural Language Processing (NLP) and Speech Technologies through tasks proposed by the Italian research community. It provides a common framework for addressing open linguistic challenges and real-world applications, with increasing attention to multilingual and multimodal settings. The 2026 edition included 10 tasks and attracted 54 participating groups from 13 countries, confirming the growing international interest in the initiative. The workshop presents the results of the evaluation campaign, highlighting the widespread adoption of Large Language Models and the organizers' effort in designing challenging tasks aimed at meaningfully evaluating and stress-testing such models.

Keywords

NLP, Evaluation, Italian, Language Models

1. Introduction

The Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA) is the biennial initiative aimed at promoting the development of language and speech technologies for the Italian language. EVALITA is promoted by the Italian Association of Computational Linguistics (AILC)¹ and it is endorsed by the Italian Association for Artificial Intelligence (AIXIA)² and the Italian Association for Speech Sciences (AISV)³.

EVALITA provides a shared framework where different systems and approaches can be scientifically evaluated and compared across a diverse set of tasks, proposed and structured by the Italian research community. The tasks are designed to represent significant scientific challenges, allowing methods, resources, and systems to be tested against shared benchmarks that capture both linguistic open issues and real-world applications. In the current NLP landscape, increasingly shaped by the advent of Large Language Models (LLMs), the organization of these tasks offers a unique opportunity to investigate how traditional and modern approaches perform in a rapidly evolving domain. The collected datasets continue to provide valuable resources for researchers to explore both established and emerging

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ cutugno@unina.it (F. Cutugno); alessio.miaschi@ilc.cnr.it (A. Miaschi); a.palmeroaprosio@unitn.it (A. P. Aprosio); giulia.RAMBELLI@univ-amu.fr (G. Rambelli); lucia.siciliani@uniba.it (L. Siciliani); marcoantonio.stranisci@unito.it (M. A. Stranisci)

id 0000-0001-9457-6243 (F. Cutugno); 0000-0002-0736-5411 (A. Miaschi); 0000-0002-1484-0882 (A. P. Aprosio); 0000-0003-3379-6941 (G. Rambelli); 0000-0002-1438-280X (L. Siciliani); 0000-0001-9337-7250 (M. A. Stranisci)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<http://www.ai-lc.it>

²<http://www.aixia.it>

³<http://www.aisv.it>

problems in Italian NLP, fostering the development of innovative solutions and enabling a community-wide discussion on the evolving capabilities and limitations of modern language technologies. While some tasks remain a core element of the evaluation campaign, new challenges are regularly introduced to reflect the changing research priorities and technological advances in the field.

This paper introduces the tasks proposed at EVALITA 2026 and provides an overview of the participants and systems whose descriptions and obtained results are reported in these Proceedings. The EVALITA 2026 final workshop, held in Bari on February 26-27th, counts 10 different tasks. In particular, the selected tasks are grouped into four research areas (tracks) according to their objective and characteristics, namely: (i) *Authorship Analysis*; (ii) *Computational Ethics*; (iii) *Emerging Tasks*; (iv) *Multimodality*; (v) *New Challenges in Long-Standing Tasks*; (vi) *Pragmatics*.

2. EVALITA 2026 Tracks and Tasks

In the 2026 edition of EVALITA, 11 different tasks were proposed, peer-reviewed, and accepted. Subsequently, one organizing team withdrew, and as a result, 10 tasks were ultimately presented at EVALITA. Data were produced by the task organizers and made available to the participants. For the future availability of this data, we are going to release them on GitHub⁴, in accordance with the terms and conditions of the respective data sources. Such a repository will also reference alternative repositories managed by the task organizers. The tasks of EVALITA 2026 are grouped according to the following tracks corresponding to five broad research areas:

Authorship Analysis

DeSegMa-It: DEtection and SEgmentation of MACHine generated texts in Italian [1]. DeSegMa-It is the first shared task aimed at testing the robustness of machine-generated text (MGT) detectors by evaluating their performance under settings where the IID (i.e. independent and identically distributed) assumption does not hold. In particular, the shared task was organized around two subtasks: i) document-level detection of MGTs and ii) human-machine text segmentation.

Computational Ethics

GSI:detect – Detecting Gender Stereotypes in Italian [2]. GSI:detect is the shared task for the recognition and classification of gender stereotypes (GSs). The task is organized into two subtasks: i) GS detection, in which systems have to assign to a text a GS value and ii) GS classification, in which systems are required to assign a score to each input text from a set of six predefined categories (e.g. role, relational, etc.).

MultiPRIDE: Multilingual Automatic Detection of Reclamation of Slurs in the LGBTQ+ Context [3]. MultiPRIDE is the shared task focused on the detection of slur reclamation, a linguistic phenomenon in which derogatory terms are reappropriated by members of a community. The task is divided into two main tasks: i) binary detection of reclamation in social media posts and ii) contextual detection of reclamation considering users' biographies. Each task is further divided into language-specific subtasks (Italian, Spanish and English).

Emerging Tasks

Cruciverb-IT: Crossword Solving [4]. Cruciverb-IT is the first shared task on Italian crossword solving. The task comprises two subtasks: 1) *clue answering*, aimed at answering individual crossword clues given the expected answer length, and 2) *grid solving*, focused on autonomously solving complete crossword grids of varying sizes.

⁴<https://github.com/evalita>.

SVELA: Selective Verification of Erasure from LLM Answers [5]. SVELA is the shared task on machine unlearning verification in Large Language Models. Participants must determine whether specific fictional identities are retained, forgotten (targeted by unlearning), or never seen during training. The evaluation is multilingual (Italian, Spanish, French, German) and tests both entity-level and instance-level detection.

Multimodality

EVWSD-ITA: Enhanced Visual Word Sense Disambiguation for Italian [6]. EVWSD-ITA is the shared task on Visual Word Sense Disambiguation (VWSD). Specifically, the task involves selecting the most appropriate image for a target word within a given context, specifically designed to include "hard negatives" (i.e. co-hyponyms that share a common hyperonym but represent distinct concepts).

New Challenges in Long-Standing Tasks

ATE-IT: Automatic Term Extraction – Italian Testbed [7]. ATE-IT is the shared task on Automatic Term Extraction (ATE) for Italian, focused on the domain of municipal waste management. The task is organized into two subtasks: i) *Term extraction*, aiming at identifying domain-specific terms in instructional texts and ii) *Term Variants Clustering*, focusing on grouping morphological and semantic term variants.

PFB: Prometeia Financial Benchmark – Benchmarking Language Models in the Financial Domain [8]. The Prometeia Financial Benchmark task evaluates how well large and small language models understand financial-domain content. It introduces a multilingual benchmark of about 1,000 multiple-choice questions, each with a correct answer and difficulty level. Participants are asked to build or adapt systems to answer these domain-specific MCQs. Two tracks are available: (i) an Italian-only corpus; (ii) a multilingual corpus (Italian, English, Turkish).

Pragmatics

FadeIT: Fallacy Detection in Italian Social Media Texts [9]. FadeIT is the first shared task on fallacy detection in social media texts in Italian. The shared task is articulated into two subtasks: i) *post-level fallacy detection*, aiming at predicting the fallacy types expressed in each input post, and ii) *span-level fallacy detection*, aiming at predicting all text segments expressing any given fallacy type in each input post.

IMPOLS: IMplicit contents in POLitical Speech [10]. IMPOLS is the shared task on automatic detection and classification of implicit contents in political speeches. The task is divided into three subtasks: i) implicit content detection; ii) implicit classification and iii) implicature type classification.

3. Participation

EVALITA 2026 attracted the interest of a large number of researchers from academia and industry, for a total of 54 single teams composed of about 127 individuals participating in one or more of the 10 proposed tasks. After the evaluation period, 57 system descriptions were submitted (reported in these proceedings), i.e., a 11.8% increase with respect to the previous EVALITA edition [11]. Moreover, task organizers allowed participants to submit more than one system result (called runs), for a total of 274 submitted runs.

Overall, EVALITA 2026 involved participants from 13 different countries. As expected, Italy was the most represented country, with a total of 44 teams having at least one Italian affiliation. Figure 1 shows the geographical distribution of participating teams excluding Italy, highlighting the international reach of the campaign. Contributions originated from several regions worldwide, with Vietnam being the most represented foreign country, accounting for 5 teams with at least one affiliated author. This

Track	Task	Teams	Runs
<i>Authorship Analysis</i>	DeSegMa-It	6	21
<i>Computational Ethics</i>	GSI:detect	6	91
	MultiPRIDE	18	83
<i>Emerging Tasks</i>	Cruciverb-IT	5	17
	SVELA	4	14
<i>Multimodality</i>	EVWSD-ITA	1	3
<i>New Challenges in Long-Standing Tasks</i>	ATE-IT	9	13
	PFB	2	4
<i>Pragmatics</i>	FadelT	7	25
	IMPOLS	2	3

Table 1

Number of participating teams and number of runs organized by track and task. The data reported is an overestimation with respect to the systems described in the proceedings (e.g. teams participating in more than a task are counted according to the number of tasks they participated in).

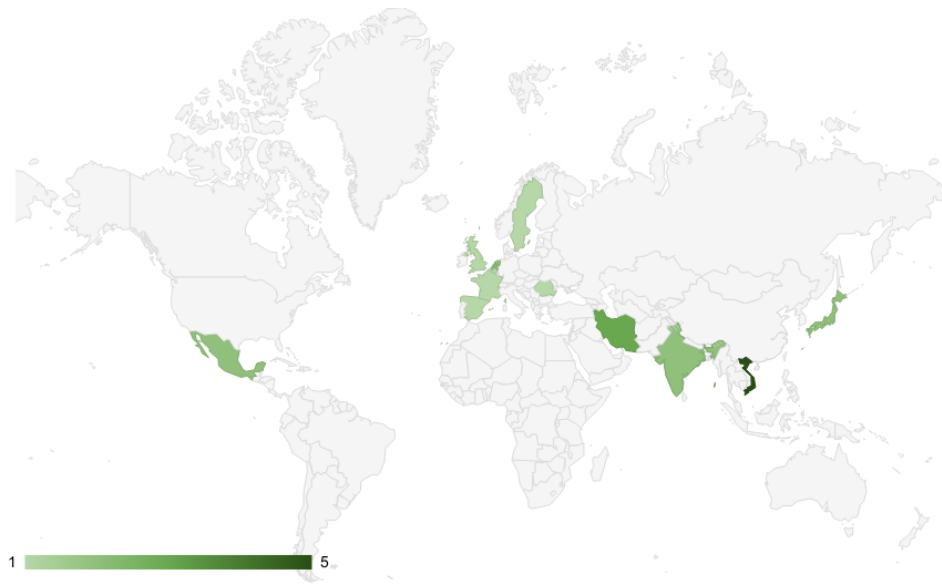


Figure 1: Number of teams per country (excluding Italy).

is followed by Iran, and by India, Japan, Mexico, and the Netherlands with 2 teams each, while all remaining countries are represented by a single team. Among the proposed tasks, MultiPRIDE stands out as the one attracting the largest international participation, with 8 teams from eight different non-Italian countries. This is followed by Fade-IT and GSI:detect, which also show a notable level of international engagement. From an institutional perspective, EVALITA 2026 involved a total of 35 affiliations, including 29 academic institutions and 6 industrial organizations, further confirming the predominantly academic nature of the participation, with a non-negligible contribution from industry.

Table 1 shows the different tracks and tasks along with the number of participating teams and submitted runs. The data reported in the table is based on information provided by the task organizers at the end of the evaluation process. Such data represents an overestimation with respect to the systems described in the proceedings. Participation was quite imbalanced across different tracks and tasks, as reported in Figure 2: each rectangle represents a task whose size reflects the number of participants, while the color indicates the corresponding track. The *Computational Ethics* track emerges as the most attended overall, with MultiPRIDE and GSI:detect standing out as the two most participated tasks of the campaign. While MultiPRIDE attracted the largest number of teams, GSI:detect resulted in a higher number of submitted runs, indicating a particularly strong engagement from participating groups. This interest can be linked to the societal relevance of the addressed phenomena, as well as to

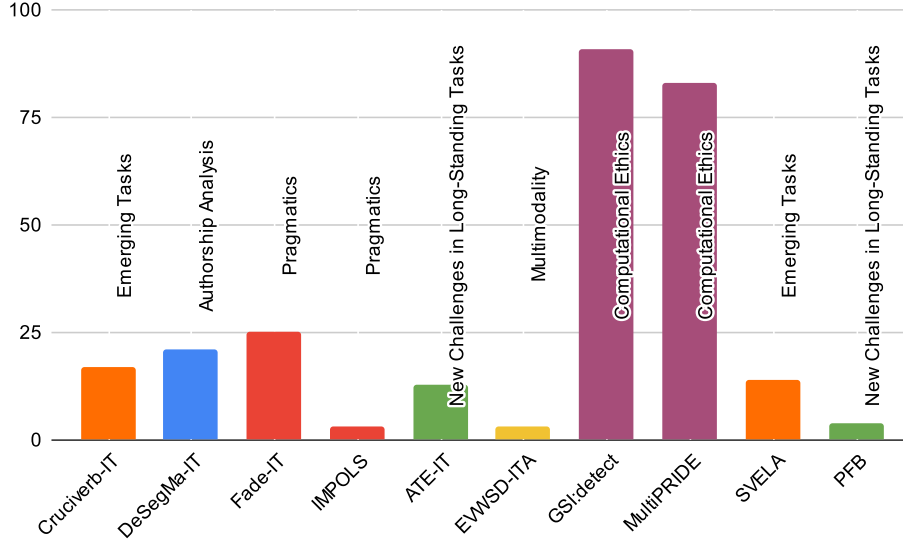


Figure 2: Number of participating teams organized by track (color) and task.

the multilingual and classification-oriented nature of the tasks. A substantial level of participation is observed in the Pragmatics track, where FadeIT attracted a relatively large number of teams, reflecting the strong interest in challenges related to fallacy detection and the analysis of pragmatic phenomena in social media. In the same track, IMPOLS shows a more limited participation, possibly reflecting both the higher complexity of modeling implicit meaning in political discourse and the additional challenges posed by its multimodal setting, which involves the joint processing of speech audio and textual transcriptions. The Authorship Analysis track, which in this edition includes DeSegMa-It, still exhibits a solid level of participation, confirming the relevance of research on machine-generated text detection within the Italian NLP community. Among the *Emerging Tasks*, Cruciverb-IT and SVELA exhibit different participation patterns. Cruciverb-IT attracted a moderate number of teams, likely due to the novelty and specificity of crossword solving in Italian, while SVELA reflects a growing interest in machine unlearning and verification in Large Language Models. The *Multimodality* track, represented by EVWSD-ITA, shows a more contained participation, consistent with the higher entry barriers typically associated with multimodal evaluation settings. Finally, tasks belonging to the *New Challenges in Long-Standing Tasks* track display heterogeneous participation. ATE-IT continues to attract a solid number of teams, confirming sustained interest in automatic term extraction, whereas PFB, despite its industrial relevance and multilingual design, involves a smaller number of participants, likely due to the domain-specific nature of the benchmark. In line with the effort to foster a more international and inclusive evaluation campaign, task organizers were encouraged to propose multilingual tasks, providing datasets in Italian and in one or more additional languages. In the 2026 edition, two tasks explicitly adopted a multilingual setting, namely MultiPRIDE, which includes Italian, Spanish and English, and PFB, which covers Italian, English and Turkish. Although the number of multilingual tasks is still limited, this represents a further step towards strengthening the international profile of EVALITA and broadening its impact beyond the Italian-speaking research community.

Turning to the types of systems submitted by participants, several interesting patterns emerge from the analysis. As shown in Figure 3, decoder-only approaches are not the most widely adopted, despite their prominence in recent NLP research. Encoder-only architectures remain the predominant choice, while a non-negligible number of teams also rely on non-LLM-based, symbolic, and encoder-decoder approaches. This distribution indicates that participants did not simply apply Large Language Models in inference-only settings, but instead explored task-specific modeling strategies aimed at building effective systems. In turn, this reflects the fact that the proposed tasks were, on average, sufficiently challenging and, in many cases, not readily solvable by directly applying Large Language Models without additional

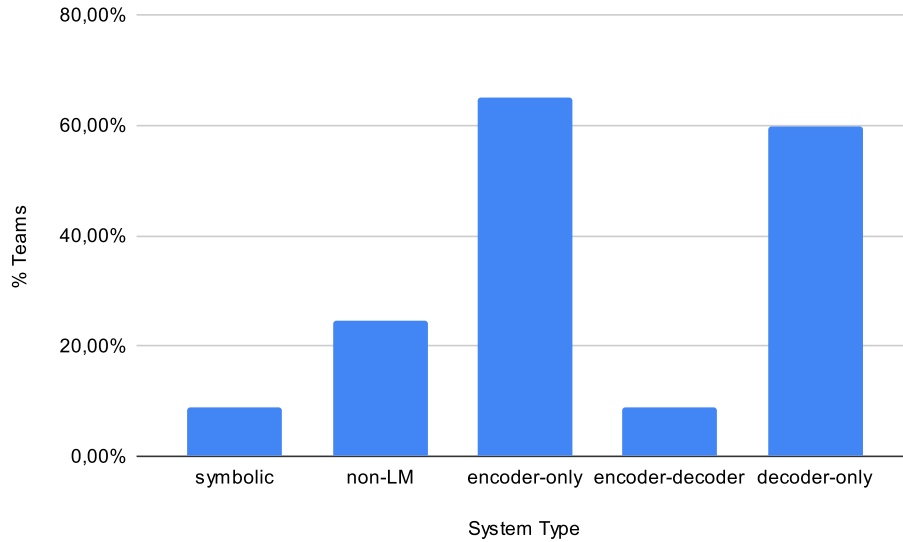


Figure 3: Percentage of the systems submitted to EVALITA 2026 according to the five categories of: symbolic, non-LM, encoder-only, encoder-decoder, decoder-only.

modeling choices. In this respect, it is worth noting that several teams still adopted traditional machine learning pipelines based on feature extraction, often leveraging linguistically motivated or task-specific features, rather than relying exclusively on end-to-end neural architectures. Moreover, a substantial number of submissions made use of data augmentation strategies, frequently involving the generation of synthetic data to enrich the training material. This practice represents an emerging trend in the campaign and is in line with recent work exploring the use of synthetic data to improve robustness and generalization in NLP systems.

As a final remark, we highlight the case of a team that participated in multiple tasks (i.e. DeSegMa, MultiPRIDE, IMPOLS and FadelT), which submitted systems for four different tasks while adopting a unified modeling approach. In particular, the team relied on the same core architecture across all submissions, leveraging a BabyLM model pre-trained on a dedicated Italian dataset.

4. “Federico Sangati” Best System Across Tasks Awards

In line with the previous edition, we confirmed the award for the best system across-task. Starting from this year, the award is dedicated to the memory of Federico Sangati. The award was introduced with the aim of fostering student participation in the evaluation campaign and in the workshop. A committee of 4 members (Dominique Brunato, Camilla Casula, Alessandro Mazzei and Marco Polignano) was asked to choose the best system across tasks. The composition of the committee is balanced with respect to the level of seniority as well as to their academic background (computer science-oriented vs. humanities-oriented). In order to select a short list of candidates, the task organizers were invited to propose candidate system participating in their tasks during the reviewing process (not necessarily top-ranking). The committee was provided with the list of candidate systems and the criteria for eligibility, based on:

- *novelty* with respect to the state of the art;
- *originality*, in terms of identification of new linguistic resources, identification of linguistically motivated features, and implementation of a theoretical framework grounded in linguistics;
- *critical insight*, paving the way to future challenges (deep error analysis, discussion on the limits of the proposed system, discussion of the inherent challenges of the task);
- *technical soundness* and *methodological rigor*.

We collected 13 system nominations from the organizers of 9 tasks from across all tracks. The candidate systems are authored by 45 authors, among whom 24 are students, either at the master’s or PhD level. The award recipient(s) will be announced during the final EVALITA workshop, during the “Awards Session” plenary session.

5. Final Remarks

The outcomes of the EVALITA 2026 campaign provide a timely snapshot of how the Italian NLP community is currently positioning itself with respect to the rapid evolution of language technologies. While Large Language Models continue to play a central role in contemporary research, an important finding of this edition is that decoder-only architectures are not the dominant modeling choice among participating systems. Instead, encoder-only models remain the most widely adopted approach, and a substantial number of teams still rely on encoder-decoder, non-LLM-based and symbolic pipelines. This distribution clearly indicates that, rather than simply applying large generative models in inference-only settings, participants favored task-oriented modeling strategies tailored to the specific linguistic and computational challenges posed by each task. This trend highlights an important shift with respect to recent editions: the proposed tasks appear, on average, not to be directly solvable by off-the-shelf large language models without additional modeling choices. As a consequence, many submissions combine neural architectures with task-specific components and linguistically motivated features, and several systems still follow more traditional machine learning pipelines. In this sense, EVALITA 2026 confirms the continued relevance of hybrid and feature-based approaches, especially when dealing with fine-grained phenomena such as fallacy detection, implicit content classification, gender stereotypes, and term variation.

Another emerging pattern concerns the increasing use of data augmentation strategies, often based on the generation of synthetic training material. A non-negligible number of systems exploit automatically generated data to compensate for limited annotated resources and to improve robustness and generalization. This practice reflects a growing interest in exploiting generative models as data producers rather than solely as end-to-end predictors, and represents a promising direction for future editions of the campaign.

From a thematic perspective, participation across tracks remains heterogeneous. The Computational Ethics track stands out as the most attended, with MultiPRIDE and GSI:detect attracting the largest number of teams and runs. This result confirms the strong interest of the community in socially relevant NLP applications, such as stereotype recognition and the analysis of reclaimed slurs in online communication. A similarly high level of engagement is observed for tasks dealing with machine-generated text and fallacies in social media, which directly address pressing challenges related to content authenticity and online discourse. In contrast, tasks requiring deeper modeling of implicit meaning or multimodal reasoning show lower participation, likely reflecting both the higher conceptual complexity of the phenomena and the higher technical barriers associated with these settings. EVALITA 2026 also strengthens the role of the campaign as a testbed for novel and emerging research directions. The introduction of tasks such as Italian crossword solving and selective verification of unlearning in LLMs demonstrates a growing interest in moving beyond traditional classification benchmarks, towards evaluation scenarios that explicitly target reasoning, memorization, and controllability of language models.

With respect to the international dimension, the 2026 edition confirms the growing involvement of research groups from outside Italy, with contributions from thirteen different countries. Although the majority of teams still include at least one Italian affiliation, the geographical distribution of participants and the strong international participation in some tasks indicate a steadily increasing visibility of the EVALITA initiative within the broader NLP community.

Overall, the results of EVALITA 2026 underline a mature and diversified research landscape, in which LLMs coexist with more traditional approaches, and where linguistic insight, task-specific modeling and evaluation-driven design continue to play a crucial role. The campaign confirms the importance

of shared evaluation initiatives as a means to foster methodological rigor, stimulate the exploration of emerging challenges, and support the development of robust and socially responsible language technologies for Italian and beyond.

Acknowledgments

We would like to thank our sponsors: AI2B⁵, Aptus.ai⁶ and Prometeia⁷. Our gratitude goes also to the University of Bari Aldo Moro for hosting the event. In addition, we sincerely thank the Best System award committee for providing their expertise and experience. Moreover, we acknowledge the AILC Board members for their trust and support. We warmly thank our invited speaker, Paolo Rosso, for sharing his knowledge and insights with his talk. Last but not least, we would like to thank all the task organizers and participants who made this edition special with their enthusiasm and creativity.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5 and Grammarly to conduct grammar and spelling checking. The author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] G. Puccetti, A. Pedrotti, A. Esuli, DeSegMa-IT at EVALITA 2026: Overview of the "Detection and Segmentation of Machine generated texts in Italian" Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini, GSI:detect at EVALITA 2026: Overview of the Task on Detecting Gender Stereotypes in Italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [3] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, MultiPRIDE at EVALITA 2026: Overview of the Multilingual Automatic Detection of Slur Reclamation in the LGBTQ+ Context Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [4] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, M. Nissim, Cruciverb-IT @ EVALITA 2026: Overview of the Crossword Solving in Italian Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [5] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, SVELA at EVALITA 2026: Overview of the Selective Verification of Erasure from LLM Answers Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [6] E. Musacchio, L. Siciliani, P. Basile, G. Semeraro, EVWSD-ITA at EVALITA 2026: Overview of the Enhanced Visual Word Sense Disambiguation for Italian Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [7] N. Cirillo, G. M. Di Nunzio, F. Vezzani, ATE-IT at EVALITA 2026: Overview of the Automatic Term Extraction Italian Testbed Task, in: Proceedings of the Ninth Evaluation Campaign of Natural

⁵<https://www.ai2b.it/>

⁶<https://aptus.ai/it/>

⁷<https://www.prometeia.com/it>

Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.

- [8] A. P. Bardelli, T. Çekiç, I. Demirtaş, M. Filannino, S. Scala, A. Galassi, G. Pappacoda, P. Torrioni, PFB at EVALITA 2026: Overview of the Prometeia Financial Benchmark, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [9] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the Fallacy Detection in Italian Social Media Texts Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [10] L. Gregori, W. Paci, V. Saccone, IMPOLS at EVALITA 2026: Overview of the IMPOLS task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), Bari, Italy, 2026.
- [11] M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, Evalita 2023: Overview of the 8th evaluation campaign of natural language processing and speech tools for italian, in: EVALITA Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian Final Workshop: Parma, Italy, September 7-8th, 2023, Accademia University Press, 2024, p. 3.

Tu chiamale se vuoi emozioni: The impact of emotions on disinformation and sexism identification

Paolo Rosso¹

¹Universitat Politècnica de València, Spain

1. Abstract

Harmful information in social media, both disinformation and hate speech, is pervasive. Emotional content can be exploited by malicious agents to provoke inflammatory responses and exacerbate polarisation in online debate. Disinformation often is conveyed through text with a high emotional valence. Emotions have in fact a key role: fake news tend to trigger different emotions and of different intensity to the users compared to texts containing truthful information. Readers prefer high arousal news, which catches more attention than regular news and sticks easily on mind. Moreover, the emotional bias may distort the perception of argumentation, for instance bringing one person to refute to acknowledge facts that upset them. In this talk I will mainly focus on disinformation and how this problem is perceived and addressed at EU level, being disinformation campaigns (often from abroad) considered as a form of information warfare and an attempt to trigger negative emotions and create polarization in our democracies. I will also show what kind of emotional responses may trigger narratives where conspiracy theories are mentioned. The language of these narratives is inherently different, and it works on a subconscious, emotional layer, exploiting the readers' biases that may distort the perception of argumentation, for instance bringing one person to refute to acknowledge facts that upset them. In the last part of the talk, I will address the problem of sexism identification in memes, showing what kind of emotions are triggered and how emotional signals can be extracted with physiological data collected via eye-tracking, heart rate, and EEG signals.

2. Short Bio

Paolo Rosso studied Computer Science in Italy at the University of Pisa, did his PhD in Ireland at the Trinity College Dublin, and ended up in Spain at the Technical University of Valencia where he's Full Professor and a member of the Pattern Recognition and Human Language Technology (PRHLT) Research Center and the Valencian Graduate School and Research Network of Artificial Intelligence (ValgrAI). He was advisor of 30 PhD students and at the moment he is of seven. He published more than 400 peer-reviewed papers in conferences and journals. During the last years, he was the PI of several research projects, such as FairTransNLP_Stereotypes: Fairness and Transparency for equitable NLP applications in social media-Identifying stereotypes and prejudices and developing equitable systems, FAKEnHATE-PoC: FAKE news and HATE speech, and XAI-DisInfodemics: eXplainable AI for disinformation and conspiracy detection during infodemics, funded by MCIN/AEI and by European Union NextGenerationEU/PRTR. His research interests focus on AI for social media related tasks such as the detection of harmful information, both fake news and hate speech. He collaborated with the Spanish National Security Department and with the Science and Technology Office (Oficina-C) of the Spanish Congress of Deputies in topics related to disinformation campaigns and AI, and with the Spanish Observatory on racism and xenophobia (OBERAXE). He helped in the organisation of more than 50 shared tasks in evaluation forums such as SemEval, CLEF, FIRE, IberEval/Iberlef and of course Evalita

Evalita 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ proso@dsc.upv.es (P. Rosso)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

(MultiPRIDE, AMI, SardiStance, IronITA, SENTIPOLC). At the moment he is actively involved in the analysis of physiological signals (EEG variables, heart rate, blinks, fixations, saccades) collected during the viewing of sexist memes and videos. He currently serves as Associate Editor at IEEE Transactions on Affective Computing.

Personal Website. <https://personales.upv.es/prosso/>

Kenji-Endo: a BabyLM @EVALITA

Calogero Jerik Scozzaro¹, Mattero Rinaldi¹, Gianluca Mittone¹ and Marco Antonio Stranisci^{1,2}

¹University of Torino, Computer Science Department, Corso Svizzera 185, 10149 Torino, Italy

²aequa-tech, Turin, Italy

Abstract

We present Kenji-Endo, a BabyLM pretrained on a dedicated Italian dataset, which participated in four tasks at the 9th edition of EVALITA: DeSegMa, MultiPRIDE, IMPOLS, and FadeIT. Kenji-Endo achieved competitive performance across all tasks, demonstrating that language modeling with limited data and compact model sizes can represent a viable alternative to Large Language Models.

Keywords

NLP, Evaluation, Italian, Language Models, BabyLM

Katsumata: Senti, diventeresti mio amico?

Kenji-Endo: Mi va anche bene, però diventare amici non è una cosa che si decide, sai?

Naoki Urasawa

1. Introduction

Large Language Models (LLMs) have become a dominant paradigm in Natural Language Processing and their adoption has gone beyond the academic field, supporting a high number of real-life applications. For a long time, the development of these models followed the scaling law principle [1], according to which bigger models and datasets would have led to better performance. LLMs dramatically increased in the number of parameters: BERT [2] and GPT-2 [3], both released in 2019, counted 110M and 124M parameters; Llama4 Behemoth, released in 2025, 2T parameters [4]. This growth has also been facilitated by architectural innovations such as the Mixture-of-Experts (MoE) paradigm [5, 6]. In MoE models, the standard feed-forward layers are replaced by multiple parallel “expert” networks, and a routing mechanism dynamically selects a small subset of them for each input token. As a result, only a fraction of the total parameters is activated at each forward pass, allowing the model to scale to very large parameter counts while keeping computational costs relatively limited. The same happened for datasets’ size, which scaled from billions [7] to trillions of tokens [8].

Alongside this significant growth, new approaches are emerging as critical alternatives focused on environmental impact [9] and more cognitively inspired architectures [10]. Among these approaches, the BabyLM challenge [11, 12, 13] is emerging as a novel line of research in sample-efficient language modeling. The initiative calls for small architectures that aim to simulate developmental learning processes by limiting the amount of pretraining data to developmentally plausible quantities, typically enforcing budgets of 10M and 100M words of text-only data. In addition to data constraints, the challenge also encourages limited training durations, further emphasizing efficiency in learning. Initially conceived as an English-based project, BabyLM models have been proposed for Italian [14], bilingual [15], and multilingual [16] settings.

In this paper we present Kenji-Endo: a BabyLM system presented at the 9th edition of EVALITA [17]. Whilst inspired by BabyLM, the focus on Kenji-Endo is not on developmental learning but on testing

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ calogerojerik.scozzaro@unito.it (C. J. Scozzaro); matteo.rinaldi@unito.it (M. Rinaldi); gianluca.mittone@unito.it (G. Mittone); marcoantonio.stranisci@unito.it (M. A. Stranisci)

🆔 0009-0003-8311-7801 (C. J. Scozzaro); 0000-0001-9457-6243 (M. Rinaldi); 0000-0001-9457-6243 (G. Mittone); 0000-0001-9457-6243 (M. A. Stranisci)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Group	Tokens
Books	18.1M
Conversations	3.4M
Forums	13.2M
Laws	5.5M
News	18.4M
Social Media	273K
Subtitles	4.3M
Wikipedia	13.8M

Table 1

Composition of the Kenji-Endo corpus by text group, source, and total number of characters.

the ability of a small architecture against larger LLMs. To this extent, we developed the Kenji-Endo dataset, a corpus of approximately 135 million tokens collected from different sources (e.g., legal texts, subtitles) and filtered using a text readability criterion. This dataset was used to pretrain Kenji-Endo from scratch. We adopted three pretraining settings: two standard dense transformer architecture and a Mixture-of-Experts (MoE) variant.

Kenji Endo has been tested on the following EVALITA tasks: DeSegMa-IT [18], MultiPRIDE [19], IMPOLS, and FadeIT [20]. Results obtained by our system show that Kenji-Endo achieves competitive performance across multiple EVALITA tasks, including the second-best accuracy on DeSegMa-IT Subtask A and the best score on IMPOLS Subtask 1, despite its small size and limited pretraining computation. However, performance degrades in settings that require the exploitation of noisy or highly abstract contextual information, highlighting limitations in handling complex pragmatic phenomena. Despite these limitations, Kenji-Endo demonstrates that small models can remain effective and competitive when carefully pretrained and adapted to downstream tasks.¹

2. System Description

In this section, we describe the architecture and training pipeline of the proposed system. We first introduce the Kenji-Endo corpus developed for pretraining, detailing its composition and data selection criteria. We then describe the pretraining setup and the resulting pretrained models, followed by the fine-tuning and prompt-based strategies employed to adapt the models to the downstream classification tasks.

2.1. The Kenji-Endo Corpus

In order to pretrain our system, we developed the Kenji-Endo corpus, a dedicated dataset composed of documents selected from different sources and belonging to different text types. Particular attention was paid to choosing datasets that may be small in terms of token count but capable of providing a diverse and peculiar data mixture. Given that the object of the work is a BabyLM, CHILDEES, a corpus focused on children’s language, was included.² We also included Kiparla [21] and VoLip [22] as examples of spoken text transcriptions, as well as [23] and [24], two corpora derived from tweets. Parts of Eurlex and Gazzetta Ufficiale [25] were also incorporated. We included a dump of articles from DueParole,³ given its focus on easy-to-read Italian. News, blog posts and discussion board posts were obtained from TestiMole [26], alongside public domain books sourced mainly from LiberLiber⁴ and the clean Wikipedia dump contained in TestiMole. A synthetic description of the Kenji-Endo corpus is reported in Table 1.

¹Code and data of our system are available at <https://github.com/neurosymbolic-unito/kenji-endo-at-evalita>

²The corpus was obtained from the Italian language section of <https://talkbank.org/childes/access/Romance/>

³Articles accessible at <http://dueparole.it>

⁴<https://liberliber.it/>, included in TestiMole

An additional criterion adopted in the construction of the Kenji-Endo dataset was a filtering process based on text readability. To this extent, we exploited *Il Nuovo Vocabolario di Base della Lingua Italiana*⁵, a dictionary of high-frequency Italian words developed by Tullio De Mauro. We used the vocabulary to classify documents according to their readability by computing the relative frequency of high-occurring terms within each document. For each document type, we computed the third-quartile threshold of this distribution and retained only documents whose values exceeded this threshold, corresponding to texts with a higher presence of high-frequency terms. The final corpus is composed of 184,000 documents and 83.80 millions of tokens.

2.2. Pre-Training Setup

The BabyLM pre-training is run on a single NVIDIA Grace Hopper Superchip (1x Grace A02 CPU 72 cores + 1x NVIDIA H100 GPU) on the HPC4AI research cluster hosted by the University of Turin, Italy [27]. The training code is based on PyTorch exploiting the xFLL framework, which allows high-performance, effortless deployment of AI training on distributed infrastructures [28]. From a technical perspective, the whole computation exploits the half-precision bfloat16 data format to reduce GPU RAM impact, halving the memory required to store the model’s parameters, activations, and gradients, while improving the overall computational performance, since the GH100 can process such low-precision data more than one order of magnitude faster than standard FP32 [29]. Activation checkpointing is also adopted to save additional GPU RAM and thus allowing even larger batch sizes, which empirically provide better learning convergence; the final training batch size used is 64 samples. AdamW is used as optimizer, as common in contemporary practice, with a starting learning rate of $2e^{-4}$, weight decay of 0.1, and betas of (0.9, 0.95). A warmup-cosine-decay learning rate scheduler is adopted to obtain improved and more stable learning convergence, using 1% of the available batches for the warmup phase, and then decreasing the learning rate to 1% of its initial value over a single training epoch (1,272 batches in total). During pre-training, we used a maximum context window of 1024 tokens and trained the models for a total of 10 epochs. Gradient clipping is set to 1.0, and no further gradient accumulation is used.

This pre-training procedure resulted in two pretrained language models, differing in their architectural design:

- KENJI-ENDO VANILLA⁶: a dense transformer-based causal language model with 12 layers and a hidden size of 768, trained using full causal self-attention at every layer and based on the Qwen3ForCausalLM architecture. This model follows a standard decoder-only architecture and has approximately 130M parameters.
- KENJI-ENDO MoE⁷: a Mixture-of-Experts (MoE) variant with the same number of layers and hidden dimensionality as the vanilla model, in which the standard feed-forward blocks are replaced by sparse expert routing, based on the Qwen2MoeForCausalLM architecture. At each token, a single expert is selected among 7 available experts, together with one shared expert that is always active, enabling increased model capacity while maintaining a comparable computational cost, for a total of approximately 240M parameters.

Additionally, we pretrained a further model variant employing more recent architectural features: KENJI-ENDO ALiBi.⁸ This model was trained using the Matformer⁹ framework instead of the Hugging Face Trainer. Matformer is a new in-house developed library that focuses on better modularity and code readability, while providing readily available support for different configurations and optimization strategies. KENJI-ENDO ALiBi was trained for 8 hours on a single Nvidia 3090 GPU connected to a Thinkpad R61 laptop, an extremely economical setting that highlights the low access barrier needed for

⁵<https://github.com/marcostranisci/babyLM/blob/main/resources/nvdb.csv>

⁶<https://huggingface.co/CCC-Unito/kenji-endo-1.0-vanilla>

⁷<https://huggingface.co/CCC-Unito/kenji-endo-1.0-moe>

⁸<https://huggingface.co/CCC-Unito/kenji-endo-1.0>

⁹<https://github.com/mrinaldi97/matformer>

developing BabyLMs. It consists of 12 layers, a hidden size of 768, a SwiGLU feed-forward network with an intermediate size set to three times the hidden size, and uses RMSNorm for layer normalization. The model was trained for 10 epochs with a batch size of 16 and a gradient accumulation of 12, adopting a warmup-hold-cosine decay lr scheduling that peaked at $1e-3$ and decayed until $8e-5$. As a positional embedding strategy, AliBi [30] was used instead of RoPe [31] to enable generalization and scaling beyond the 1024 token sequence length employed during training. Flash attention 2, Liger Kernels and Nvidia’s transformer-engine are natively supported by the Matformer library and were used to speed up training. Finally, Muon [32] was used instead of AdamW as optimizer, with a Nesterov-style momentum of 0.95 and 5 Newton–Schulz iteration steps. Compared to AdamW, Muon allows for faster convergence, an aspect that may be crucial in the development of Baby-LMs. The final model achieved performance comparable to that of KENJI-ENDO VANILLA; however we reserve for the future to conduct more tests especially to compare the performances in long context scenario.

2.3. Fine-Tuning and Prompt-Tuning Setup

The pretrained models are trained with next-token prediction objective. However, the tasks addressed in this work are framed as classification problems, which require an additional adaptation step in order to map the model outputs to discrete class labels. To this end, we explored two classification strategies on the KENJI-ENDO VANILLA and KENJI-ENDO MoE models: a discriminative fine-tuning approach and a generative, prompt-based approach. In the discriminative setting, we fine-tuned the models using a batch size of 8 and the `AutoModelForSequenceClassification`¹⁰ framework, which connects the pretrained model to a classification head. The classification head consists of a linear layer applied to the hidden representation of the last token of the input sentence, producing class logits. The model is trained by directly optimizing a cross-entropy loss over the class labels. All model parameters were updated during fine-tuning. This configuration was applied only to the vanilla model, as the MoE architecture employed does not support this class. Training was carried out for 3 epochs on a single NVIDIA T4 GPU. In the generative setting, no fine-tuning was performed. Instead, we adopted a prompt-based approach applicable to both vanilla and MoE models, casting the classification task as text generation. We used a batch size of 4 and a simple prompt that includes the input text followed by a class label request (e.g., “*Testo: {text} Classe:*”).

For all experiments, we fixed the maximum sequence length to 1024 tokens, including tasks in which a portion of the instances exceeded this length. For each task, we split the original training set into 80% training and 20% validation data. This split was used to fine-tune and select the best checkpoint in the discriminative setting. In the generative setting, no parameter updates were performed, as the models were evaluated in a zero-shot fashion. The validation split was used solely for configuration comparison and model selection. During fine-tuning, model checkpoints were saved at each epoch, and the checkpoint achieving the best performance on the validation set was selected. Unless otherwise specified, we relied on the default hyperparameter settings provided by the Hugging Face Trainer.¹¹ Fine-tuning was carried out on NVIDIA A40 GPUs available on the HPC4AI infrastructure. The compute time of each model has been 1110.70 seconds on average.

3. Results Overview

In this section, we present an overview of the tasks in which our team participated and report the corresponding experimental results. For each task, we provide a brief description and compare our performance against other participating systems using the official evaluation metrics. Table 2 summarizes the final configurations adopted for each subtask.

¹⁰https://huggingface.co/transformers/v3.0.2/model_doc/auto.html#automodelforsequenceclassification

¹¹https://huggingface.co/docs/transformers/main_classes/trainer

Task	Subtask	Strategy	Model
DeSegMa	A	Discriminative	KENJI-ENDO VANILLA
MultiPRIDE	A	Discriminative	KENJI-ENDO VANILLA
MultiPRIDE	B	Discriminative	KENJI-ENDO VANILLA
IMPOLS	1	Discriminative	KENJI-ENDO VANILLA
IMPOLS	2	Generative	KENJI-ENDO VANILLA
FadeIT	A	Discriminative	KENJI-ENDO ALIBI

Table 2

Overview of the tasks and subtasks in which Kenji-Endo participated, including the strategy and model variant used for the final submission.

Position	Team	Accuracy
1	Gradient Descenders	0.9458
2	Kenji Endo	0.9426
3	UniTor	0.9288
4	Nicla	0.9243
5	Stochastic Gradient Descenders	0.9216

Table 3

Results of DeSegMa-IT Subtask A ranked by accuracy.

3.1. DeSegMa

DeSegMa-IT (Detection and Segmentation of Machine Generated Text in Italian) [18] focuses on the detection of machine-generated text in Italian. It is composed of two subtasks: Subtask A, where given an input text, systems are required to perform binary classification, distinguishing between human-written and machine-generated content; and Subtask B, where participants are required to detect the boundary between human-written text and its machine-generated continuation by identifying the character index marking the beginning of the machine-generated content. Our participation was limited to Subtask A.

Due to the large size of the DeSegMa-IT training set, under the discriminative strategy we explored two training configurations: *i*) training for a single epoch on the full dataset, and *ii*) training for three epochs on a reduced subset of 3,000 instances. Overall, we evaluated four experimental configurations: *i*) vanilla discriminative training on the full dataset, *ii*) vanilla discriminative training for three epochs on the reduced subset, *iii*) vanilla generative inference, and *iv*) MoE generative inference. Based on validation performance, the vanilla discriminative model trained for a single epoch on the full dataset achieved the best results (Accuracy: 0.9952, F1-score: 0.9952) and was therefore selected for the final submission.

Table 3 reports the official results on the DeSegMa-IT Subtask A test set. Our system achieved the second-best performance among the participating teams, with an accuracy of 0.9426 and, ranking closely behind the top-performing approach. A closer inspection of the classification results shows a generally balanced performance across the two classes. Overall precision, recall, and F1-score are comparable (Precision: 0.9455, Recall: 0.9426, F1-score: 0.9426), with only minor differences in class-wise metrics. For the human-written class, recall is higher (0.9824) than precision (0.91), while for the machine-generated class, precision (0.98) is higher than recall (0.9031). This behavior shows a cautious tendency in assigning the machine-generated label, characterized by higher precision than recall. For completeness, we conducted a full analysis by evaluating the performance of all four experimental configurations. The submitted configuration achieved the best overall performance (F1-score: 0.9426). The discriminative configuration trained for three epochs on the reduced subset showed slightly lower results (F1-score: 0.9207), suggesting that exposure to the full training data was more beneficial than additional epochs on a smaller subset. The generative approaches, using the vanilla and MoE models, achieved competitive but lower scores (F1-scores of 0.9207 and 0.9225, respectively), indicating that the prompt-based formulation did not match the performance of full discriminative fine-tuning on this task.

3.2. MultiPRIDE

MultiPRIDE is a binary classification task in which systems are required to determine whether a term related to an LGBTQ+ context in a sentence is used with a reclamatory intent. The task is structured into two subtasks: Subtask A, where only the textual content of the tweet is provided; and Subtask B, where, in addition to the tweet content, participants have access to optional contextual information related to the author’s profile, such as their biography, when available. The task is proposed in a multilingual setting, encompassing data in Italian, Spanish, and English. Our team participated in Subtasks A and B for the Italian language.

For each subtask, we evaluated three experimental configurations, corresponding to the vanilla discriminative setting and the two generative inference setups described in Section 2.3. In both cases, the discriminative KENJI-ENDO VANILLA model achieved the best performance on the development set and was therefore selected for the final submission. Specifically, for Subtask A, the selected model obtained a development set Accuracy of 0.9174 and an F1-score of 0.7429, while for Subtask B it achieved a development set Accuracy of 0.8394 and an F1-score of 0.6316.

Table 4 reports the official rankings for Subtasks A and B on the Italian language, evaluated in terms of macro F1-score. Our system achieved a macro F1-score of 0.8360 in Subtask A and 0.7489 in Subtask B, placing in the mid-to-lower range of participating systems.

Analyzing results, it is possible to observe that the model’s performance is skewed towards precision in the Subtask A while the opposite happens for Subtask B. Besides being detrimental in the overall performance, the additional context appears to produce a greater number of false positive (+0.13) than false negatives (+0.2). To further investigate this behavior, we analyzed the tokenization characteristics of the inputs. We observed that the biography field contains a substantially higher proportion of byte-fallback tokens (e.g., emojis and flag symbols) than the tweet text. Specifically, the mean byte-fallback ratio is 0.1149 for biographies compared to 0.0134 for tweet text, and 34.3% of biography entries contain at least one such token (versus 9.9% for tweet text). This suggests that the additional contextual information in Subtask B may introduce noisy and semantically fragmented token sequences, which can dilute the useful signal and provide a plausible explanation for the lower F1-score despite the richer input.

3.3. IMPOLS

IMPOLS focuses on the automatic recognition of implicit content in Italian political speech. Given an utterance within its context, systems are required to detect and classify implicit, non-bona fide contents, i.e., questionable information that is not explicitly stated but is implicitly conveyed and understood as true. Such content is commonly used in political communication to convey potentially manipulative messages implicitly. The task is structured into three subtasks: 1) a binary detection task aimed at identifying the presence of questionable implicit content; 2) a binary classification task distinguishing between implicatures and presuppositions; and 3) a multiclass classification task focusing on the categorization of implicatures into particularized conversational, generalized conversational, or conventional. Our team participated in Subtasks 1 and 2. For each subtask, we evaluated multiple experimental configurations, similarly to the previous tasks. For Subtask 1, the discriminative KENJI-ENDO VANILLA model achieved the best performance on the development set and was therefore selected for the final submission. Specifically, the selected model obtained a development set accuracy of 0.9507 and an F1-score of 0.9521. For Subtask 2, the generative KENJI-ENDO VANILLA model achieved a development set accuracy of 0.5027 and an F1-score of 0.4417.

Table 5 reports the official results obtained on the IMPOLS test set. Our system achieved the best performance in Subtask 1, obtaining an F1-score of 0.9496 and ranking first among the participating teams. From a closer look at model’s performance two main aspects emerge: *i*. Most of the errors are false negatives (63%), showing that the model lacks of recall; *ii*. The model struggles with discourses from a specific speaker - Pierluigi Bersani - whose speech excerpts represent 29.2% of all the wrong prediction. The second speaker with the highest number of wrong predictions is Mario Draghi (4, 9.7%).

Pos.	Team (Run)	F1
1	Ghavidel-Rajabi (1)	0.8981
2	MilaNLP (1)	0.8959
3	Ghavidel-Rajabi (2)	0.8909
4	SaFe Tweets (1)	0.8895
5	LlaNa (1)	0.8835
6	GRUPPETTOZZO (1)	0.8834
7	Challenger (1)	0.8816
8	HateItOff (1)	0.8809
9	GRUPPETTOZZO (2)	0.8735
10	baseline (1)	0.8731
11	UniBO-FICLIT (1)	0.8707
12	NamDang (2)	0.8589
13	HateItOff (2)	0.8584
14	NetGuardAI (1)	0.8537
15	The Hate Busters (2)	0.8503
16	I2C (2)	0.8435
17	NamDang (1)	0.8407
18	Kenji-Endo (1)	0.8360
19	The Hate Busters (1)	0.8345
20	MilaNLP (2)	0.8306
21	KIT-TIP-NLP (2)	0.8103
22	Avahi (1)	0.7904
23	KIT-TIP-NLP (1)	0.7659
24	I2C (1)	0.6202

(a) Subtask A

Pos.	Team (Run)	F1
1	LlaNa (1)	0.9021
2	baseline (1)	0.8981
3	GRUPPETTOZZO (1)	0.8979
4	The Hate Busters (2)	0.8827
5	MilaNLP (1)	0.8827
6	GRUPPETTOZZO (2)	0.8681
7	UniBO-FICLIT (1)	0.8644
8	MilaNLP (2)	0.8641
9	AIWizards (1)	0.8564
10	AIWizards (2)	0.8549
11	HateItOff (2)	0.8462
12	The Hate Busters (1)	0.8324
13	HateItOff (1)	0.8319
14	SaFe Tweets (1)	0.8164
15	Avahi (1)	0.8031
16	DataSummit (1)	0.7734
17	Kenji-Endo (1)	0.7489
18	NetGuardAI (1)	0.7405

(b) Subtask B

Table 4

Official rankings for MultiPRIDE (Italian) in terms of macro F1-score for Subtasks A (left) and B (right). The number in parentheses indicates the submitted run identifier.

Team	Subtask 1	Subtask 2	Subtask 3
kenji-endo	0.9496	0.2226	–
RES2	0.9089	0.8762	0.6369
Baseline	0.4602	0.3352	0.2942

Table 5

Results on the IMPOLS task in terms of F1-score across the three subtasks.

Performance on Subtask 2 was substantially lower (F1-score of 0.2226). An inspection of the model predictions revealed a degenerate behavior, with the model predicting a single class, which negatively impacted the overall performance on this subtask. The drop of performance might suggest the lack of model’s ability to distinguish between more abstract concepts related to pragmatics, due to the limited amount of pretraining data that does not include discipline-specific knowledge.

3.4. FadeIT

FadeIT is a task focused on the detection of fallacies expressed in Italian social media texts and is based on the FAINA dataset [33]. AINA consists of Italian social media posts manually annotated by expert annotators for the presence of argumentative fallacies, with over 11,000 span-level annotations spanning 20 different fallacy types (e.g., ad hominem, strawman). The criterion for selecting the texts was their belonging to three specific categories: migration, climate change and public health. Each text can be classified as expressing zero, one, or multiple fallacy types. The annotations also account for human label variation, highlighting the intrinsic subjectivity of fallacy identification. The task is structured into two subtasks: Subtask A, “post-level fallacy detection”, is framed as a multi-label classification tasks in which the model should correctly detect the fallacies expressed in a given text.

Position	Team (Run)	P	R	F1
1	TiGRO (3)	53.24	59.99	56.39
2	MALTO (2)	55.75	53.60	54.63
3	TiGRO (2)	53.43	51.48	52.41
4	UNICA (1)	55.22	45.23	49.71
5	TiGRO (1)	62.52	39.85	48.65
6	UNICA (3)	51.19	44.26	47.45
7	Kenji-Endo (1)	49.38	45.55	47.37
8	UNICA (2)	58.70	38.44	46.44
9	MALTO (1)	46.56	43.65	45.05
10	RBG-AI (2)	33.09	57.78	42.07
11	RBG-AI (3)	30.65	57.62	40.00
12	Label (3)	27.60	68.08	39.26
13	Label (1)	52.82	30.94	38.96
14	RBG-AI (1)	36.35	41.11	38.57
15	Label (2)	52.76	30.11	38.32
16	Baseline	38.53	14.28	20.84

Table 6

Test set results for FadeIT Subtask A. Systems are ranked by decreasing micro F1-score. The number in parentheses indicates the submitted run identifier.

Subtask B, “fine-grained fallacy detection”, instead, is framed as a multilabel span classification task in which the goal is to detect the specific text segments in which the fallacy is expressed. Our team participated in subtask A. For this purpose, the decoder head of the autoregressive model was replaced with a linear classification head, taking as input the output of a mean pooling applied to the models’ last layer. To stabilize training and avoid overfitting, we set dropout at 0.5.

Results of Table 6 show that our system ranked 7 out of 15 runs (16 including the baseline) with a micro F1-score of 47.37, with a precision of 49.38 and a recall of 45.55. Compared to the baseline (F1: 20.84), this corresponds to a substantial improvement of +26.53 points. The finer-grained analysis of results shows important limitations in our system, though. Kenji-Endo was not able to identify 14 fallacy-types out of 19 confirming the issues encountered in other tasks where models have to deal with more complex phenomena.

4. Conclusion and Future Work

In this paper, we presented Kenji-Endo, a BabyLM system that participated in four EVALITA tasks: DeSegMa, MultiPRIDE, IMPOLS, and FadeIT. The system was pretrained from scratch on a dedicated corpus using two architectures (simple auto-regressive and Mixture of Experts). Across tasks, Kenji-Endo achieved competitive results, ranking second in DeSegMa Subtask A and first in IMPOLS Subtask 1, while obtaining solid mid-range performance in MultiPRIDE and FadeIT. However, it systematically struggles with context-dependent tasks, as it fails to distinguish pragmatics concepts (implicatures *versus* presupposition) and to handle noisy autobiographical signals such as emojis. Despite the existing limitation, Kenji-Endo demonstrates that small models can remain effective and competitive when carefully pretrained and adapted to downstream tasks. Future work will proceed along two main directions. We will investigate methods to improve performance on more complex and abstract linguistic phenomena, such as the detection of cultural and intertextual references, which remain challenging for models at this scale. We will systematically study the interaction between data recipes and model learning by conducting extensive ablation experiments, with the goal of disentangling the effects of training procedures from those of data composition on downstream task performance.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (OpenAI) and Grammarly in order to: Improve writing style and Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, arXiv preprint arXiv:2001.08361 (2020).
- [2] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.
- [3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, OpenAI blog 1 (2019) 9.
- [4] A. Meta, The llama 4 herd: The beginning of a new era of natively multimodal ai innovation, <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>, checked on 4 (2025) 2025.
- [5] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, J. Dean, Outrageously large neural networks: The sparsely-gated mixture-of-experts layer, in: International Conference on Learning Representations, 2017.
- [6] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, et al., Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models, arXiv preprint arXiv:2401.06066 (2024).
- [7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, Journal of machine learning research 21 (2020) 1–67.
- [8] L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, et al., Dolma: An open corpus of three trillion tokens for language model pretraining research, in: Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: long papers), 2024, pp. 15725–15788.
- [9] M. C. Rillig, M. Ågerstrand, M. Bi, K. A. Gould, U. Sauerland, Risks and benefits of large language models for the environment, Environmental science & technology 57 (2023) 3464–3466.
- [10] I. Lee, T. Berg-Kirkpatrick, Readability $\neq$ learnability: Rethinking the role of simplicity in training small language models, in: Second Conference on Language Modeling, ????
- [11] A. Warstadt, A. Mueller, L. Choshen, E. Wilcox, C. Zhuang, J. Ciro, R. Mosquera, B. Paranjabe, A. Williams, T. Linzen, et al., Findings of the babylm challenge: Sample-efficient pretraining on developmentally plausible corpora, in: Proceedings of the BabyLM challenge at the 27th conference on computational natural language learning, 2023, pp. 1–34.
- [12] M. Y. Hu, A. Mueller, C. Ross, A. Williams, T. Linzen, C. Zhuang, R. Cotterell, L. Choshen, A. Warstadt, E. G. Wilcox, Findings of the second babylm challenge: Sample-efficient pretraining on developmentally plausible corpora, in: The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning, 2024, p. 1.
- [13] L. G. G. Charpentier, L. Choshen, R. Cotterell, M. O. Gul, M. Y. Hu, J. Liu, J. Jumelet, T. Linzen, A. Mueller, C. Ross, et al., Proceedings of the first babylm workshop, in: Proceedings of the First BabyLM Workshop, 2025.
- [14] L. Capone, A. Suozzi, G. E. Leboni, A. Lenci, Bambi goes to school: Evaluating italian babylms with invalsi-ita (2025).
- [15] Z. Shen, A. Joshi, R.-C. Chen, Bambino-lm:(bilingual-) human-inspired continual pre-training of

- babylm, in: Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, 2024, pp. 1–7.
- [16] J. Jumelet, A. Fourtassi, A. Haga, B. Bunzeck, B. Shandilya, D. Galvan-Sosa, F. G. Haznitrana, F. Padovani, F. Meyer, H. Hu, et al., Babybabellm: A multilingual benchmark of developmentally plausible training data, arXiv preprint arXiv:2510.10159 (2025).
 - [17] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [18] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, 2026.
 - [19] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [20] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [21] E. Gorla, C. Mauri, Il corpus KIParla: una nuova risorsa per lo studio dell’italiano parlato, volume 2, CLUB- Circolo linguistico dell’Università di Bologna, Bologna, 2018, pp. 96–116. URL: <https://hdl.handle.net/2318/1689340>.
 - [22] M. Voghera, C. Iacobini, F. Cutugno, R. Savy, I. Alfano, A. De Rosa, Il VoLIP: una risorsa per lo studio della variazione nel parlato della lingua italiana, in: Actes du XXVIIe Congrès international de linguistique et de philologie romanes, ELiPhi, Strasbourg, 2016, pp. 1651–1663.
 - [23] M. A. Stranisci, S. Frenda, M. Lai, O. Araque, A. T. Cignarella, V. Basile, C. Bosco, V. Patti, O-dang! the ontology of dangerous speech messages, in: I. Kernerman, S. Carvalho, C. A. Iglesias, R. Sprugnoli (Eds.), Proceedings of the 2nd Workshop on Sentiment Analysis and Linguistic Linked Data, European Language Resources Association, Marseille, France, 2022, pp. 2–8. URL: <https://aclanthology.org/2022.salld-1.2/>.
 - [24] A. Ramponi, C. Casula, DiatopIt: A corpus of social media posts for the study of diatopic language variation in Italy, in: Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 187–199. URL: <https://aclanthology.org/2023.vardial-1.19>.
 - [25] E. Federici, M. Ferraretto, N. Landro, Gazzetta Ufficiale: A dataset of legislative texts, public and private acts, 2024. URL: <https://huggingface.co/datasets/mii-llm/gazzetta-ufficiale>.
 - [26] M. Rinaldi, R. Varvara, V. Patti, Testimole-conversational: A 30-billion-word italian discussion board corpus (1996-2024) for language modeling and sociolinguistic research, 2026. URL: <https://arxiv.org/abs/2602.14819>. arXiv:2602.14819.
 - [27] M. Aldinucci, S. Rabellino, M. Pironti, F. Spiga, P. Viviani, M. Drocco, M. Guerzoni, G. Boella, M. Mellia, P. Margara, I. Drago, R. Marturano, G. Marchetto, E. Piccolo, S. Bagnasco, S. Lusso, S. Vallero, G. Attardi, A. Barchiesi, A. Colla, F. Galeazzi, HPC4AI: an ai-on-demand federated platform endeavour, in: D. R. Kaeli, M. Pericàs (Eds.), Proceedings of the 15th ACM International Conference on Computing Frontiers, CF 2018, Ischia, Italy, May 08-10, 2018, ACM, 2018, pp. 279–286. URL: <https://doi.org/10.1145/3203217.3205340>. doi:10.1145/3203217.3205340.
 - [28] I. Colonnelli, R. Birke, G. Malenza, G. Mittone, A. Mulone, J. Galjaard, L. Y. Chen, S. Bassini, G. Scipione, J. Martinovič, V. Vondrák, M. Aldinucci, Cross-Facility Federated Learning, Procedia Computer Science 240 (2024) 3–12. doi:10.1016/j.procs.2024.07.003.
 - [29] NVIDIA Developer, NVIDIA Hopper Architecture In Depth, 2023. URL: <https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/>, accessed: 2026-01-07.
 - [30] O. Press, N. A. Smith, M. Lewis, Train short, test long: Attention with linear biases enables input

- length extrapolation, arXiv preprint arXiv:2108.12409 (2022). URL: <https://arxiv.org/abs/2108.12409>. arXiv:2108.12409.
- [31] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, Y. Liu, Roformer: Enhanced transformer with rotary position embedding, *Neurocomputing* 568 (2024) 127063.
 - [32] K. Jordan, Y. Jin, V. Boza, Y. Jiacheng, F. Cesista, L. Newhouse, J. Bernstein, Muon: An optimizer for hidden layers in neural networks, 2024. URL: <https://kellerjordan.github.io/posts/muon/>.
 - [33] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.

DeSegMa-IT at EVALITA 2026: Overview of the "Detection and Segmentation of Machine Generated Text in Italian" Task

Giovanni Puccetti^{1,*}, Andrea Pedrotti^{1,†} and Andrea Esuli¹

¹Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" (CNR-ISTI), Via G. Moruzzi 1, Pisa, 56124, Italy

Abstract

DeSegMa-IT's shared tasks aim to test the robustness of machine-generated text (MGT) detectors by evaluating their performance under settings where the IID assumption does not hold. While state-of-the-art MGT detectors report high accuracy, such results often rely on unrealistic experimental settings: for example, relying on prior knowledge of the text generator, or failing to consider domain shifts and efficient fine-tuning - or post-tuning - strategies. In DeSegMa-IT, participants are challenged with two sub-tasks: (i) document-level detection of MGTs and the (ii) human-machine text segmentation. This paper describes the released dataset, discusses the systems submitted by participants, and provides an initial analysis of the obtained results.

Keywords

Machine-Generated Text Detection, Text Segmentation, Text Classification

1. Introduction and Motivation

Recent advancements in Generative AI and Large Language Models (LLMs) have led to the development of systems, such as GPT-4 [1], Claude¹, Llama-3 [2] and DeepSeek-V3 [3] among others, that can generate text that is often indistinguishable from human-written content [4].

This capability, along with the many beneficial applications of LLMs, also enables malicious actors to create Machine Generated Text (MGT) for deceptive purposes. For example, it can be used to manipulate online traffic and spread misinformation through content farms [5] or to influence human revisions of sensitive documents in critical domains, such as scientific peer review.²

Beyond malicious use, the widespread adoption of LLMs into everyday tools raises concerns related to authorship attribution, intellectual property, and the transparency of human-AI collaboration. In journalistic [5], educational [6], and governmental settings [7], the ability to distinguish between human-written and machine-generated content has become of uttermost importance to preserve the essential features of trust for such sensible domains, while also allowing for responsible AI deployment.

As a consequence, the task of machine-generated text (MGT) detection has received increasing attention throughout the years. With the rapid proliferation of AI-based assistants, the need for automatic tools to detect their outputs has become more urgent, leading to the proposal of numerous detection methods [8, 9, 10]. However most existing work focuses on English, highlighting the importance to develop MGT detection systems for other languages as well. Furthermore, the widespread availability of open-weight LLMs poses existing detection methods with the challenge of an ever-growing array of fine-tuned models, making the detection task increasingly non-IID (independent and identically distributed) and stressing their generalization capabilities to account for subtle shifts in writing style [11].

To address this gap, the community proposed shared tasks focused on the detection of MGT texts mainly focused on English or on multilingual settings [12, 13] along with the release of benchmark

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†] These authors contributed equally.

✉ giovanni.puccetti@isti.cnr.it (G. Puccetti); andrea.pedrotti@isti.cnr.it (A. Pedrotti); andrea.esuli@isti.cnr.it (A. Esuli)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹anthropic.com/claude-model-card

²aclweb.org/genai-peerreview-guidelines

Sub-Task A Example

Human Text: Viktor Orban, da quando il leghista è diventato ministro dell’Interno, si mandano segnali di apprezzamento reciproco. Un’alleanza che potrebbe portare l’Italia al fianco dei Paesi di Visegrad e che già - in occasione della discussione sulla riforma di Dublino - ha messo in difficoltà gli altri partner dell’Ue. Il primo ad allacciare rapporti era stato Salvini. Da Frosinone il ministro aveva parlato dell’Ungheria come di un paese con cui l’Italia potrà cambiare l’Europa. Entrambi, in fondo, sono dichiaratamente euroscettici. E sia Orban che il leghista ... **Label:** 0

Machine Text: Viktor Orban, dopo anni di duri scontri diplomatici, sono pronti a unire le loro forze per riscrivere l’agenda di Bruxelles. Il leader di Fratelli d’Italia Giorgia Meloni e il presidente del governo ungherese si sono visti a Vienna, in un vertice di "centrodestra, identità italiana e sovranità italiana". Dopo un incontro di ben tre ore i due hanno spiegato come si possano conciliare politicamente le due visioni d’Europa ... **Label:** 1

Figure 1: Sub-task A Example.

Sub-task B Example

Text: Il presidente del Tribunale internazionale del diritto del mare (Itlos), Vladimir Golitsyn, ha fissato *al 10 agosto la data in cui il tribunale arbitrale di Amburgo esaminerà le informazioni che l’Italia intende raccogliere in India per scagionare i Marò . Nei giorni scorsi su vari quotidiani erano uscite indiscrezioni circa la data di un eventuale incontro tra i due fucilieri di Marina e i loro avvocati e i funzionari del ministero dell’Interno indiano, che dovrebbero rilasciare a loro una sorta di "licenza" temporanea così che i due marinai possano recarsi in India.*

Target Character Index: 103

Figure 2: Sub-task B Example. Non-italic text represents the human segment, while the *italic* text denotes the continuation generated by a LLM.

datasets [14]. To extend these efforts to the Italian language, the EVALITA 2026 [15] shared task DeSegMa-IT aims to foster research on Italian MGTs by providing the research community with a benchmark dataset and a standardized evaluation framework.

2. Tasks Definitions

DeSegMa-IT³ shared task is organized into two sub-tasks: (i) MGT detection and (ii) human-machine text segmentation. The first sub-task evaluates detection accuracy at the document level and aims to test the robustness of machine-generated text detectors under realistic, non-IID conditions. While state-of-the-art MGT detectors report high accuracy, such results often rely on unrealistic experimental settings—for example, depending on prior knowledge of the text generator (e.g., [16]) or ignoring domain shifts and fine-tuning strategies. To address this, we simulate real-world non-IID conditions by generating the training and testing datasets from different LLMs.

In contrast, the segmentation sub-task focuses on identifying machine-generated text within a human-written document. For this task, both dataset splits share the same LLM generators due to its inherently higher difficulty.

2.1. Sub-task A: MGT Detection in the Wild

In **sub-task A**, we simulate the challenge posed by the ever-shifting domain of MGT detection by sampling train and test documents from two disjunct sets of generating LLMs. The task is structured as a binary-classification problem and defined as follows: “Given a piece of text t , assign it the label 0, if the text is written by a human, and 1 otherwise.” We provide an example in Figure 1.

³<https://desegma.github.io/>

Source	Prompt
Il Giornale	Sei un giornalista italiano che scrive sul giornale conservatore di destra "Il Giornale". Scrivi un articolo di giornale a partire da questo titolo: <NEWS TITLE>. Evita qualsiasi tipo di formattazione. Non generare il titolo, inizia direttamente dal corpo dell'articolo.
La Repubblica	Sei un giornalista italiano che scrive sul giornale progressista di sinistra "La Repubblica". Scrivi un articolo di giornale a partire da questo titolo: <NEWS TITLE>. Evita qualsiasi tipo di formattazione. Non generare il titolo, inizia direttamente dal corpo dell'articolo.

Table 1

Prompts used for the generations.

2.2. Sub-task B: Human - Machine Text Segmentation

In the **second sub-task**, participants are required to detect the boundary between the human-written text and the machine-generated continuation by identifying the index of the character that marks the beginning of the MGT content. Each data sample consists of a variable-length human-written prompt, always followed by a variable-length continuation produced by the model. Unlike traditional MGT detection tasks that require document-level binary classification, this sub-task focuses on segmentation: participants must pinpoint the beginning of the text generated by the LLM.

The task is defined as follows: *“Given a piece of text t , return the index of the first character that is generated by an LLM.”* To ensure a statistically robust evaluation, the length of the human-written sub-string is uniformly sampled from a range of 64 to 512 characters. This setup simulates real-world scenarios in which MGT may be inserted into otherwise human-written content. We provide an example in Figure 2.

3. Dataset

For each of the two sub-tasks, we provide participants with training and evaluation specific data. In this section, we describe the two datasets in details.

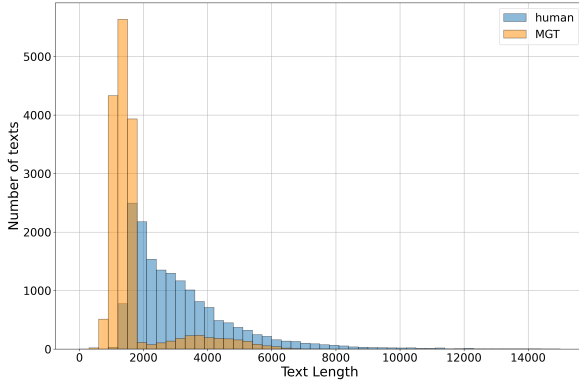
3.1. Sub-task A: MGT Detection

The dataset for sub-task A consists of both human-written and machine-generated texts. Human texts are sampled from the Change-IT dataset [17]. The original dataset consists of news articles collected from two Italian outlets: the *La Repubblica*⁴ newspaper and the *Il Giornale*⁵ newspaper. From the dataset, we retain the headline field, storing the title of the news articles and the human-written article itself. We use the headline to prompt a pool of LLMs to generate an synthetic version of the article. Furthermore, we provide the LLM with guidelines regarding the political agenda of the original news outlet. We also prompt the model to avoid any formatting style (e.g., retain from using bullet points, markdown sections, etc.) to adhere more closely to the style of the human-written articles. All models are prompted with their respective default generation parameters. We report the prompts used in Table 1. We sample a random balanced selection of human-written and machine-generated texts to construct the task’ datasets.

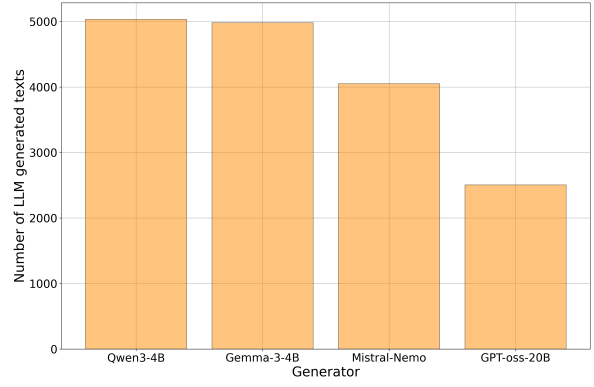
Training dataset We collect 33,138 news articles, equally split between human and machine-generated texts, while preserving a 1:1 ratio for each news outlet. The models used to create the machine generated part of the training dataset are reported in Appendix A.1.

⁴<https://www.repubblica.it/>

⁵<https://www.ilgiornale.it/>

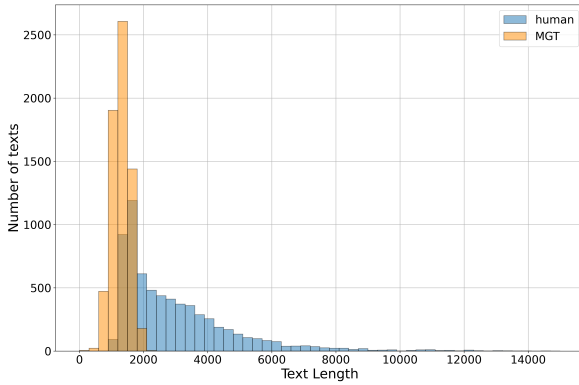


(a) Text length distribution by class (training set)

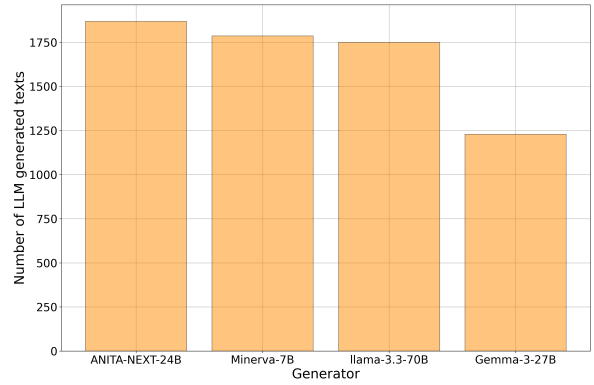


(b) Number of texts per generator (training set)

Figure 3: Text statistics for sub-task A training data.



(a) Text length distribution by class (test set)



(b) Number of texts per generator (test set)

Figure 4: Text statistics for sub-task A test data.

We sample texts from the generators to maintain a balanced distributions among LLMs. Figure 3a reports the distribution of text lengths and Figure 3b shows the generating LLMs distribution. We include 15,135 articles, balanced according to the target classes and evenly split among generators.

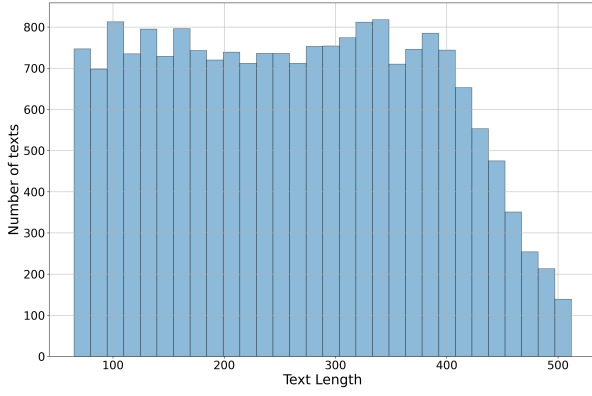
Test dataset We include 13,268 articles evenly balanced across news outlets. Figure 4a reports text length statistics and Figure 4b shows the distribution of texts with respect to the generating LLMs. Selected generators are reported in Appendix A.1.

3.2. Filtering by Perplexity

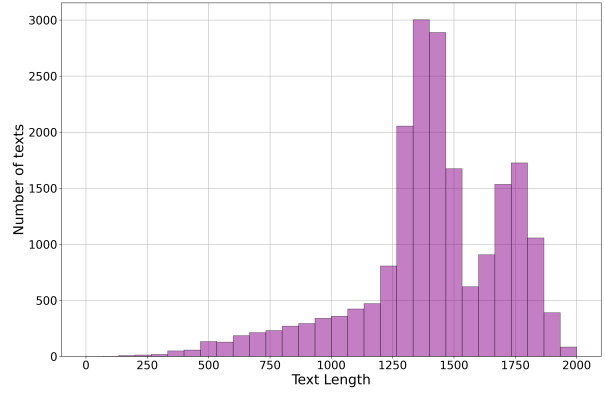
To make the classification more challenging, we compute with a held-out model, `gemma-3-12b-it`, the mean perplexity score for the human-written texts. We use this mean value to filter out MGTs that drift too far away from the selected filtering score. This process allows us to select documents that are similar in style according to the evaluator LLM, shrinking the classification boundary between the two classes.

3.3. Sub-task B: Human-MGT Segmentation

Data for the sub-task B consist of news articles switching from human written content to machine-generated continuation. Each text is paired with an integer denoting the length of the human written part. To generate plausible continuations, we select a human-written article and discard up to the first three sentences of the text. The remaining segment is used as an input prompt and fed to one of nine LLMs. The rationale for discarding the initial portion of the article is to present participants with a more

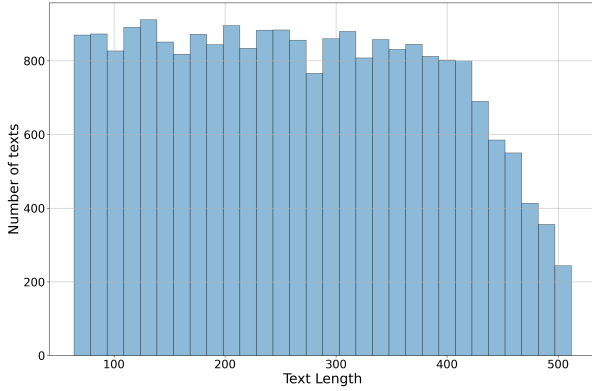


(a) Human text length distribution

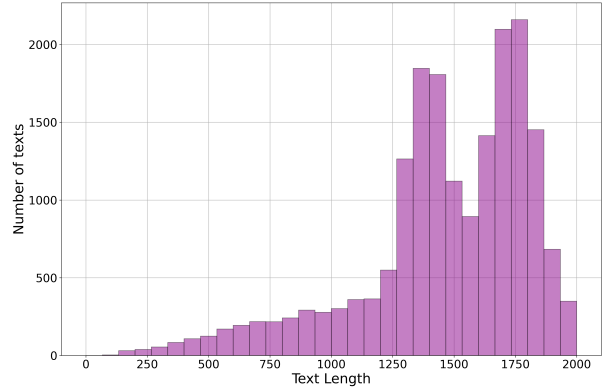


(b) Total text length distribution

Figure 5: Text length statistics for sub-task B training data.



(a) Human text length distribution (test set)



(b) Total text length distribution (test set)

Figure 6: Text length statistics for sub-task B test data.

challenging scenario, designed to mimic the detection of machine-generated text segments appearing at arbitrary positions within an article, rather than only at its beginning.

Training dataset This split consists of 19,945 news articles. Generating LLMs are reported in Appendix A.2. We set the minimum length of the human-written to 64 characters, and the maximum length to 512 characters. The length distribution of human-written segments is shown in Figure 5a and the length distribution of full human-written and machine-continued texts is reported in Figure 6b.

Test dataset For the test set, we keep the same LLMs and select 23,211 new samples, while maintaining an even distribution among models. Figure 6a shows the length distribution of the human segments at the beginning of each sample and Figure 6b the length distribution of the joined human-written and machine-continued texts.

4. Evaluation Metrics

We define the following evaluation metrics for each sub-task:

- For sub-task A: the main evaluation metric is Accuracy obtained by each system in the test set. Furthermore, we also report True Positive Rate (TPR) and False Positive Rate (FPR) for all systems.
- For sub-task B: the evaluation metric is the Mean Absolute Error (MAE) computed as follows:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - x_i|$$

Team	S	Affiliations	Sub-task		Runs	
			A	B	A	B
Gradient Descenders	3	University of Information Technology, Ho Chi Minh City, Vietnam; Vietnam National University, Ho Chi Minh City, Vietnam	✓	✓	3	1
Kenji-Endo	4	University of Turin, Turin, Italy; aequa-tech, Turin, Italy	✓		1	
MINDS	1	Politecnico di Torino, Turin, Italy		✓		2
Nicla	1	Independent	✓	✓	1	1
UniTor	5	University of Rome Tor Vergata, Rome, Italy	✓	✓	6	2
Stochastic Gradient Descenders	3	University of Information Technology, Ho Chi Minh City, Vietnam; Vietnam National University, Ho Chi Minh City, Vietnam	✓	✓	2	2
Pangram		Pangram Labs	✓		1	

Table 2

Team members, affiliations, participation, and number of submission for the test-phase. Column *S* reports the number of team members.

5. Participants

We received a total of 35 submissions from six unique teams. Each team could submit up to 10 runs per sub-task, with the best-performing one automatically selected as the final score. To handle submissions we use the Codabench platform [18].

Furthermore, the DeSegMa-IT task was joined by Pangram Labs⁶ as a non-competing industrial participant. In Table 2, we provide an overview of the participants’ affiliations, team composition, joined sub-tasks, and number of test-run submitted.

Gradient Descenders For the **sub-task A**: The team [19] employs the UmBERTo model⁷ an encoder only language model trained on Italian data. The authors add Multi Layer Perceptron with two dense layers and tanh activation as a classification head that takes the model [CLS] token as input.

For the **sub-task B**: The team approaches segmentation as a token level binary-classification task. They assign to each token a human-written or machine-generated text and train a binary classifier. At inference time, the first token in the sequence classified as human-written is selected and set as the boundary index by mapping the token boundary to its respective character. For binary classification they rely on a DeBERTa model fine-tuned for the Italian language.⁸

Kenji Endo For the **sub-task A**: The team [20] employs a decoder-only transformer architecture pre-trained from scratch on the Kenji-Endo dataset, exploring both a dense model and a Mixture-of-Experts (MoE) variant. The team investigates two classification paradigms for this sub-task: a discriminative fine-tuning approach, in which a classification head is trained on top of the dense model, and a generative, prompt-based approach, applicable to both the dense and MoE models. For discriminative training, the dense model is fine-tuned either for a single epoch on the full training set or for multiple epochs on a reduced subset, while the generative setting performs inference via prompting without additional fine-tuning. The dense discriminative model trained for a single epoch on the full dataset was selected for submission.

UniTor For the **sub-task A**: The system submitted by the team [21] consists of a fine-tuned version of ModernBERT-large trained on an augmented⁹ dataset. This augmented dataset consists of 14.000 ad-

⁶<https://www.pangram.com/>

⁷<https://github.com/musixmatchresearch/umberto>

⁸<https://huggingface.co/osiria/deberta-base-italian>

⁹Note that this was not allowed by the competition rules, as reported in the website (<https://desegma.github.io/>): "Keep in mind that you should only use the training dataset we make available to train your detectors."

ditional instances: 7,000 human-written articles from the CHANGE-IT [22] corpus of Italian newspaper articles and 7,000 synthetic samples generated by translating the English RAID benchmark [23].

For the **sub-task B**: The team leverages ModernBERT-large fine-tuned for token-level binary classification, where each token is labeled as human-authored or machine-generated. A two-layer MLP classification head produces per-token probabilities. Rather than thresholding individual token predictions, boundary localization is performed via a change point detection procedure: the boundary token is selected by maximizing a score that aggregates log-likelihood evidence for human-authored tokens before the boundary and machine-generated tokens after it.

Nicla For the **sub-task A**: Team [24] addresses the task using DistilBERT-base fine-tuned for binary text classification. The model employs a standard sequence classification head, with hyperparameters selected via Bayesian optimization on a held-out validation subset to improve accuracy and generalization.

For the **sub-task B**: The team addresses the task by training a LightGBM regressor on sentence embeddings produced by a frozen Sentence-BERT model based on all-MiniLM-L6-v2, directly predicting the character index of the human-machine boundary.

Stochastic Gradient Descenders For the **sub-task A**: Team [25] frames sub-task A as a conditional single-token generation task using a decoder-only instruction-following LLM. The team fine-tunes Qwen2.5-0.5B-Instruct via supervised instruction tuning (SFT) with a conversation-style prompt, where the model outputs "0" for human-written or "1" for machine-generated text. Low-Rank Adaptation (LoRA) is applied to all linear layers to reduce computational cost and mitigate forgetting. At inference, predictions are obtained through greedy decoding of the first token, with a fallback to inspect raw logits if the model outputs non-numeric text.

For the **sub-task B**: The team reformulates the task as a token-level sequence labeling problem, labeling each token as human-written or machine-generated. The team fine-tunes UmBERTo [26] with a standard token-level classification head on top of the encoder. During inference, the model predicts per-token labels, and the boundary is extracted as the start character of the first token classified as machine-generated. Full fine-tuning is performed with standard optimization and mixed-precision training.

MINDS For the **sub-task B**: The team [27] framed the task as a token-level sequence labeling problem, using an encoder-only transformer architecture to predict human-LLM segment boundaries. They experimented with different pretrained backbones, including BERT and RoBERTa variants, under a shared training setup, finding that an Italian-specific BERT¹⁰ model yielded the best performance. Token-level predictions are converted into hard labels via a post-training threshold selection step, where the decision threshold is tuned on a validation set.

Baseline For the **sub-task A**: The baseline system is based on a multilingual version of DeBERTa-v3 [28] with a classification head. During training the backbone model is kept frozen, and only head's parameters are updated. The baseline is trained for one epoch on the whole training set.

For the **sub-task B**: The baseline system is based on a multilingual version of DeBERTa-v3 [28] with a token-level classification head. During training the backbone model is kept frozen, and only head's parameters are updated. The baseline is trained for one epoch on the whole training set. To select the switching token, the leftmost machine-generated token is selected, and mapped to its starting character index.

¹⁰<https://huggingface.co/dbmdz/bert-base-italian-cased>

Rank	Team	Accuracy	TPR $\uparrow$	FPR $\downarrow$
1	Gradient Descenders	0.9458	0.8965	0.0048
2	Kenji-Endo	0.9426	0.9031	0.0176
3	UniTor*	0.9288	0.8621	0.0042
4	Nicla	0.9243	0.8598	0.0109
5	Stochastic Gradient Descenders	0.9216	0.8468	0.0032
	Baseline	0.8769	0.7857	0.0315
	Pangram	0.8893	0.7791	<u>0.0000</u>
	UniTor (late)	<u>0.9578</u>	<u>0.9238</u>	0.0081

Table 3

Team rankings and accuracies on sub-task A. Rows with a gray background indicate non-competing results. **Bold** scores indicate best results among competing participants, an underlined one indicates best results among both competing and non-competing systems.

	Encoder-only LLM	Decoder-only LLM	Italian LLM	Multilingual LLM	English-first LLM	Encoder-only LLM	Decoder-only LLM	Italian LLM	Multilingual LLM	English-first LLM
	Sub-task A					Sub-task B				
Gradient Descenders	✓		✓			✓		✓		
Kenji-Endo		✓	✓							
MINDS						✓		✓		
Nicla	✓				✓	✓				✓
UniTor	✓			✓		✓			✓	
Stochastic Gradient Descenders		✓		✓		✓		✓		
Baseline	✓			✓		✓			✓	

Table 4

Participants’ design choices.

6. Results

Most teams participated in both sub-tasks, with the exception of the *Kenji-Endo* team which participated only in sub-task A and the *MINDS* team which only participated in sub-task B. We describe the results of the two sub-tasks separately.

6.1. Sub-task A

Table 3 reports the results obtained by participants on sub-task A and B, respectively. Note that, for the ranking of submitted systems, we consider the accuracy score. All submitted systems outperform the baseline model. The best-performing system was submitted by the team *Gradient Descenders*, which fine-tuned the Italian only LLM *UmBERTo* with a two-layer classification head. Notably, the second-best submission, by team *Kenji-Endo*, is achieved by the only system based on a decoder-only transformer. The decoder is pre-trained from scratch on an Italian-only corpus to assess the effectiveness of smaller LM trained on curated, language-specific data. This result underscores the potential for decoder-only models to be used for document-level tasks, such as machine-generated text detection.

The two best performing participating submissions report an accuracy marginally above 0.94 with the highest being 0.9458. This indicates that while developed systems are effective at detecting machine generated texts approximately 5% of the texts are misclassified, i.e. human-written texts are classified as machine-generated or the opposite. This is a significant limitation for a sensitive task such as detecting machine generated texts. For example an error of this kind can impact school- or work-related grading or performance reviews. Table 3 also reports a late submission from the *UniTor* team (which does

Rank	Team Name	MAE ↓
1	Stochastic Gradient Descenders	52.54
2	MINDS	56.53
3	Gradient Descenders	62.66
4	UniTor	81.60
5	Nicla	102.04
	Baseline	57.56

Table 5

Team rankings and Mean Absolute Precision scores on sub-task B. Rows with a gray background indicate non-competing results.

not count for the ranking) shows that higher accuracies are possible in our dataset, in particular they achieve an accuracy of 0.9578 supporting that novel techniques can further improve the leader-board of the DeSegMa-IT task.

To provide a deeper analysis of the kind of errors made by participating teams, it is worth analyzing models’ predictions through the lens of additional metrics in addition to accuracy. For this reason, in Table 3 we report False Positive and True Positive Rates (FPR and TPR respectively).

We see that all systems have an FPR lower than 2% with the highest value 0.178 shown by *Kenji-Endo* and the lowest by *Stochastic Gradient Descenders* 0.0033. Based on these results, in the best case scenario every 1,000 human texts, 3 would be wrongly attributed to generative AI, while in the worst case this would happen for about 18 human-written texts. These results highlight that participating systems occasionally attribute human-written texts to AI systems, this is not desirable as it undermines the authorship of “authentic” content created by humans.

We also investigate which participants’ choices resulted in better performance. Specifically, Table 4 reports the main decisions made by each team when developing their system. As expected all teams used language models to tackle DeSegMa-IT, therefore we report which teams used encoder-only and/or decoder-only language models and the language each model is trained on. For sub-task A we see an almost even split between encoder- and decoder-only language models (3 encoder and 2 decoder) and interestingly the first and second best performing results are obtained by teams using an encoder-only and a decoder-only language model respectively. Concerning the model language, we see a clear benefit from using models that have been trained on Italian texts, a choice made by the two best-performing submissions.

6.1.1. Pangram Labs’ Participation

Pangram¹¹ is an online tool for the automatic detection of Machine generated texts. They adopt a transformer-based classifier trained on a dataset they develop to detect both closed source Generative AI systems such as GPT-4-0613 and open-source ones such as Llama-2-70b-chat [29]. Their approach to the DeSegMa-IT task is different from others because their system does not have access to the training set which was released at a time when the Pangram detector was already available. Due to this difference, their results are reported in Table 3 as non-competing. They have lower accuracy than models trained on the training set developed specifically for the DeSegMa-IT task, however they are the only system with 0 FPR. This result indicates that their system never attributes human-written texts to Generative AI.

6.2. Sub-task B

Table 5 reports the results for Sub-task B. Two out of the five submitted systems outperform the baseline. The best performing system was submitted by the *Stochastic Gradient Descenders* team, the team fine-tuned the Italian only LLM UmBERTo and adopted a leftmost machine-generated token decision strategy. The second-best system, by team *MINDS*, also relies on an Italian-specific variant of BERT, confirming

¹¹<https://www.pangram.com/>

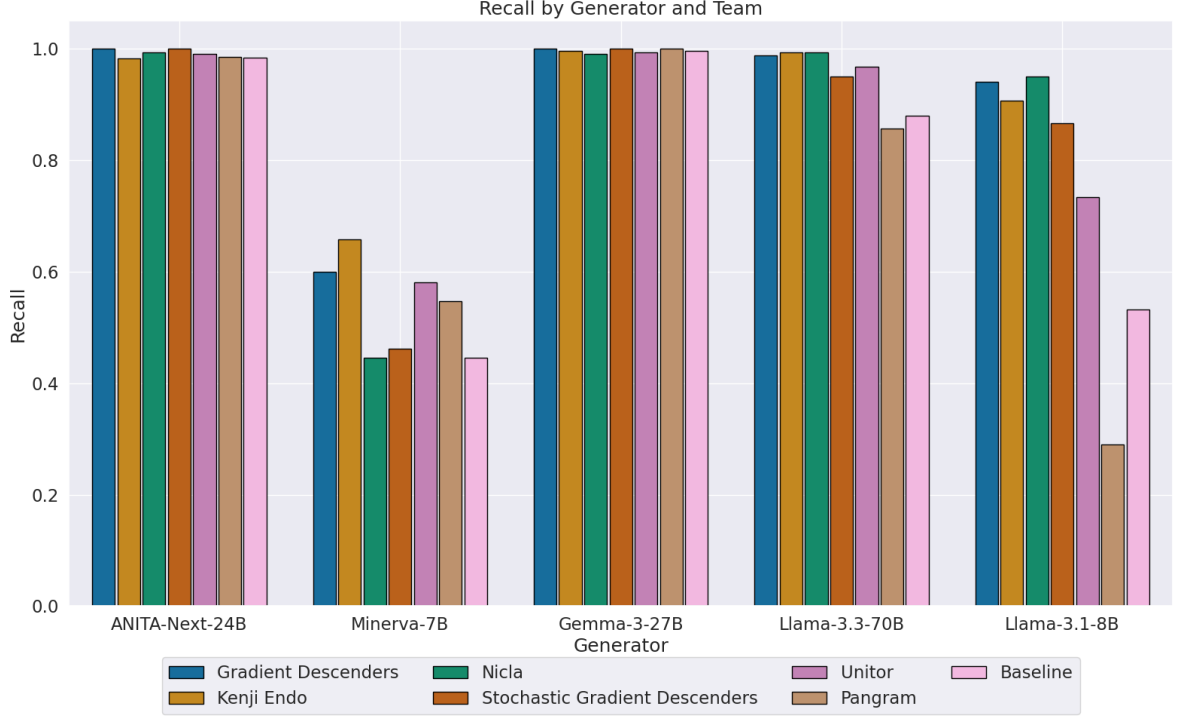


Figure 7: Sub-task A: Per-generator recall for each participant.

the relevance of language-specific pre-training for fine-grained tasks such as the human-MGT text segmentation. Among the remaining teams, the regression-based approach proposed by team *Nicla* does not achieve competitive performance, highlighting the effectiveness of framing the segmentation task as a sequence-labeling problem.

Also for sub-task B Table 4 reports the design choices, in this case there is a clear preference for encoder-only language models. Same as for sub-task A there is an advantage in using Italian-first models, the choice made by the two best-performing teams.

7. Discussion

The task of detecting machine-generated texts is a challenging task mostly due to the general purpose abilities of modern LLMs and because of the large number of available models. DeSegMa-IT’s sub-task A challenged participants to train detectors that are able to address specifically this difficulty by including synthetic texts generated by different systems. In particular, while train and test set do not share any data, they share some of the LLMs used to generate synthetic texts. As a result, the accuracy of the detectors is high but not close to a perfect score. Highlighting that sensitivity to the LLM used to generate texts is a key factor to account for when training MGT detectors.

To better understand how different generators affect detection performance, we analyze systems separately for each generator, as reported in Figure 7. This analysis is conducted exclusively on the subset of machine-generated texts, with human-authored texts excluded. As no negative instances are present in this setting, we report recall (i.e., the true positive rate) for each generator. This analysis reveals interesting patterns. All systems achieve high recall on texts generated by models based on English-first pretrained models: on texts generated by Llama-3.3-70B, and Gemma-3-27B the average recall exceeds 90%. Similar results are observed for English pretrained models later fine-tuned on Italian, ANITA-Next-24B is easy to detect, with systems maintaining a recall above 90%. In contrast, models pre-trained from scratch on Italian data are more challenging to detect, resulting in lower recall scores, close to 50%. The only model that deviates from this pattern is Llama-3.1-8B which, despite being

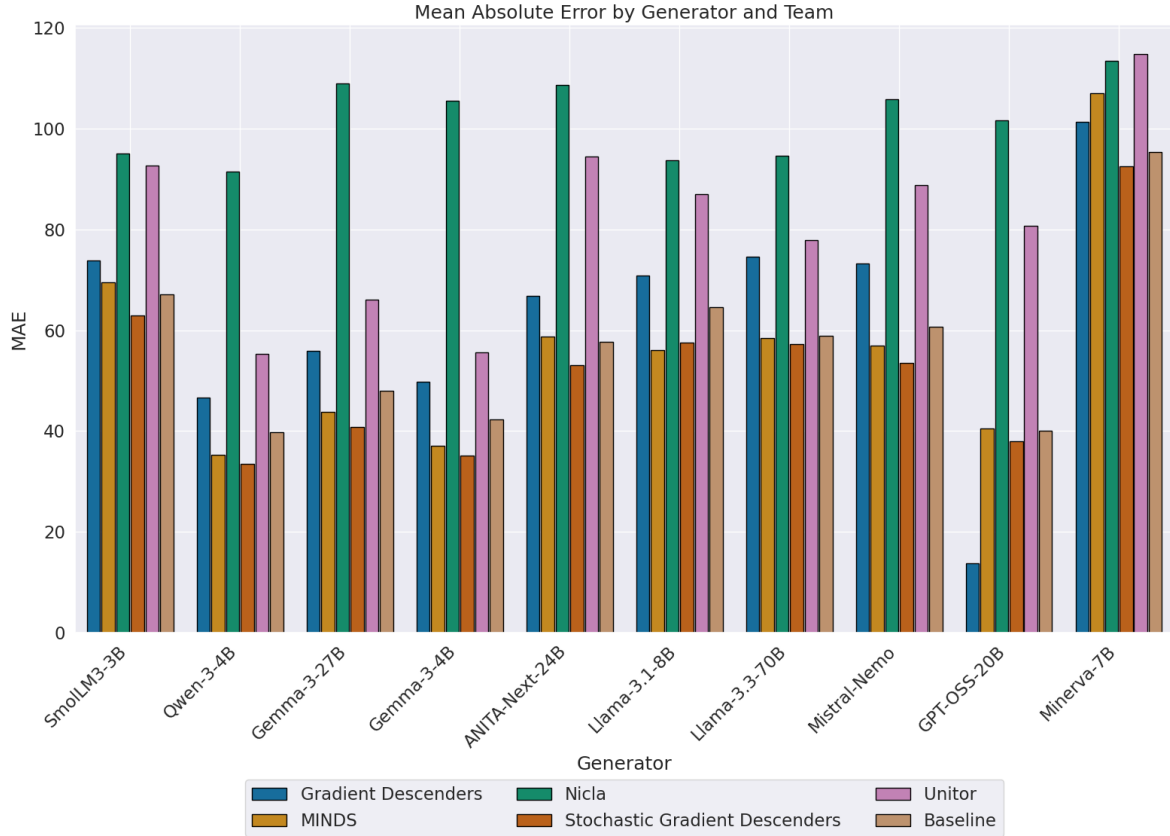


Figure 8: Sub-task B: Per-generator MAE for each participant.

pretrained on English, evades some of the detectors. We interpret this as evidence of unexpected generation patterns when the model produces Italian texts.

When performing MGT detection there are two possible errors, human-written texts classified as AI-generated and the opposite, AI-generated texts classified as human-written. In real applications the first type of error can be more harmful, since it results in genuine human effort being misattributed to AI systems. To quantify how often this happens we compute the FPR of participating systems and we find that all systems are between 0.3% and 2% FPR. While this shows that the proposed systems rarely make this type of error there are instances of human-written texts attributed to AI.

The only system showing 0 FPR is the Pangram detector which is however less accurate than others, this can be due to not relying on the DeSegMa-IT training set. Restricted to our test set, Pangram never attributes human-written texts to AI but it is more prone to attributing MGT texts to humans. This is a desirable design choice necessary for responsible deployment of MGT detectors.

Sub-task B, segmenting human-written and machine-generated texts, requires participants to identify the character where a text that is human-written in its first part switches to machine-generated. For this task, we measure performance through character-level mean absolute error, which corresponds to the number of offset characters between the predicted switch and the real one. We see that achieving an error inferior to 50 characters has proven challenging for participants, with the best system achieving a MAE of 52.54. Unlike sub-task A, identifying which models are harder to segment is not evident. Figure 8 shows the MAE achieved by each participant submission restricted to single models. There are fewer clear patterns compared to sub-task A, the main observations we draw is that SmolLM3-3B and Minerva-7B are more difficult to segment, with an average MAE above 70 characters, while GPT-OSS-20B is relatively easier with average MAE close to 40. On the remaining models, the average MAE of participants is evenly spread between 40 and 70, showing that the text generated by all models

is comparably challenging to segment from human-written text.

Acknowledgments

Giovanni Puccetti is fully funded by the Italian Ministry of University and Research under the PNRR project ITSERR (CUP B53C22001770006). Andrea Pedrotti is fully funded by the European Union - NextGenerationEU through PNRR (CUP B53C22001760006) “SoBigData.it: Strengthening the Italian RI for Social Mining and Big Data Analytics” (SoBigData.it).

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 in order to: Grammar and spelling check; Improve writing style.

References

- [1] OpenAI, GPT-4 Technical Report, 2023. URL: <https://arxiv.org/abs/2303.08774>.
- [2] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al., The llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024).
- [3] DeepSeek-AI, Deepseek-v3 technical report, 2024. URL: <https://arxiv.org/abs/2412.19437>. arXiv: 2412.19437.
- [4] L. Dugan, D. Ippolito, A. Kirubarajan, S. Shi, C. Callison-Burch, Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text, Proceedings of the AAAI Conference on Artificial Intelligence 37 (2023) 12763–12771. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/26501>. doi:10.1609/aaai.v37i11.26501.
- [5] G. Puccetti, A. Rogers, C. Alzetta, F. Dell’Orletta, A. Esuli, AI ‘news’ content farms are easy to make and hard to detect: A case study in italian, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, 2024, p. 15312–15338. URL: <http://dx.doi.org/10.18653/v1/2024.acl-long.817>. doi:10.18653/v1/2024.acl-long.817.
- [6] E. Kasneci, K. Sessler, S. Küchemann, M. Bannert, D. Dementieva, F. Fischer, U. Gasser, G. Groh, S. Günemann, E. Hüllermeier, S. Krusche, G. Kutyniok, T. Michaeli, C. Nerdel, J. Pfeffer, O. Poquet, M. Sailer, A. Schmidt, T. Seidel, M. Stadler, J. Weller, J. Kuhn, G. Kasneci, Chatgpt for good? on opportunities and challenges of large language models for education, Learning and Individual Differences 103 (2023) 102274. URL: <https://www.sciencedirect.com/science/article/pii/S1041608023000195>. doi:<https://doi.org/10.1016/j.lindif.2023.102274>.
- [7] M. Corazza, G. Longo, L. Zilli, E. Di Sante, S. Sapienza, M. Palmirani, Hybrid ai enhancing european drafting legislation for a better regulation, in: 2025 Eleventh International Conference on eDemocracy & eGovernment (ICEDEG), IEEE, 2025, pp. 106–113.
- [8] Y. Li, Q. Li, L. Cui, W. Bi, Z. Wang, L. Wang, L. Yang, S. Shi, Y. Zhang, MAGE: Machine-generated text detection in the wild, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 36–53. URL: <https://aclanthology.org/2024.acl-long.3/>. doi:10.18653/v1/2024.acl-long.3.
- [9] X. Hu, P.-Y. Chen, T.-Y. Ho, RADAR: robust ai-text detection via adversarial learning, in: Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.
- [10] M. Abassy, K. Elozeiri, A. Aziz, M. N. Ta, R. V. Tomar, B. Adhikari, S. E. D. Ahmed, Y. Wang, O. Mohammed Afzal, Z. Xie, J. Mansurov, E. Artemova, V. Mikhailov, R. Xing, J. Geng, H. Iqbal, Z. M. Mujahid, T. Mahmoud, A. Tsvigun, A. F. Aji, A. Shelmanov, N. Habash, I. Gurevych, P. Nakov, LLM-DetectAIve: a tool for fine-grained machine-generated text detection, in: D. I. Hernandez Farias,

- T. Hope, M. Li (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 336–343. URL: <https://aclanthology.org/2024.emnlp-demo.35/>. doi:10.18653/v1/2024.emnlp-demo.35.
- [11] A. Pedrotti, M. Papucci, C. Ciaccio, A. Miaschi, G. Puccetti, F. Dell’Orletta, A. Esuli, Stress-testing machine generated text detection: Shifting language models writing style to fool detectors, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 3010–3031. URL: <https://aclanthology.org/2025.findings-acl.156/>. doi:10.18653/v1/2025.findings-acl.156.
 - [12] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. M. Afzal, T. Mahmoud, G. Puccetti, T. Arnold, Semeval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection, in: A. K. Ojha, A. S. Dogruöz, H. T. Madabushi, G. D. S. Martino, S. Rosenthal, A. Rosá (Eds.), *Proceedings of the 18th International Workshop on Semantic Evaluation, SemEval@NAACL 2024*, Mexico City, Mexico, June 20-21, 2024, Association for Computational Linguistics, 2024, pp. 2057–2079. URL: <https://doi.org/10.18653/v1/2024.semeval-1.279>. doi:10.18653/v1/2024.SEMEVAL-1.279.
 - [13] Y. Wang, A. Shelmanov, J. Mansurov, A. Tsvigun, V. Mikhailov, R. Xing, Z. Xie, J. Geng, G. Puccetti, E. Artemova, J. Su, M. N. Ta, M. Abassy, K. A. Elozeiri, S. E. D. A. El Etter, M. Goloburda, T. Mahmoud, R. V. Tomar, N. Laiyk, O. Mohammed Afzal, R. Koike, M. Kaneko, A. F. Aji, N. Habash, I. Gurevych, P. Nakov, GenAI content detection task 1: English and multilingual machine-generated text detection: AI vs. human, in: F. Alam, P. Nakov, N. Habash, I. Gurevych, S. Chowdhury, A. Shelmanov, Y. Wang, E. Artemova, M. Kutlu, G. Mikros (Eds.), *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, International Conference on Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 244–261. URL: <https://aclanthology.org/2025.genaidetect-1.27/>.
 - [14] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. M. Afzal, T. Mahmoud, G. Puccetti, T. Arnold, A. F. Aji, N. Habash, I. Gurevych, P. Nakov, M4gt-bench: Evaluation benchmark for black-box machine-generated text detection, 2024. *arXiv:2402.11175*.
 - [15] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
 - [16] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: zero-shot machine-generated text detection using probability curvature, in: *Proceedings of the 40th International Conference on Machine Learning, ICML’23*, JMLR.org, 2023.
 - [17] L. D. Mattei, M. Cafagna, F. Dell’Orletta, M. Nissim, A. Gatt, CHANGE-IT @ EVALITA 2020: Change headlines, adapt news, generate, in: V. Basile, D. Croce, M. D. Maro, L. C. Passaro (Eds.), *Proceedings of Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, CEUR.org, Online, 2020.
 - [18] Z. Xu, S. Escalera, A. Pavão, M. Richard, W.-W. Tu, Q. Yao, H. Zhao, I. Guyon, Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform, *Patterns* 3 (2022) 100543. URL: <https://www.sciencedirect.com/science/article/pii/S2666389922001465>. doi:<https://doi.org/10.1016/j.patter.2022.100543>.
 - [19] T. P. T. Nhan, B. H. Son1, D. V. Thin, Gradient descenders at DeSegMa-IT: Leveraging monolingual transformer for LLM-generated text detection and boundary identification, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
 - [20] C. J. Scozzaro, M. Rinaldi, G. Mittone, M. A. Stranisci, Kenji-Endo: a BabyLM at EVALITA 2026, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
 - [21] F. Borazio, G. D. Luca, D. Pasquini, D. Croce, R. Basili, UniTor at DeSegMa-IT Analyzing supervision and encoder representations for italian machine-generated text detection, in: *Proceedings*

- of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [22] L. D. Mattei, M. Cafagna, F. Dell’Orletta, M. Nissim, A. Gatt, CHANGE-IT @ EVALITA 2020: Change headlines, adapt news, generate (short paper), in: V. Basile, D. Croce, M. D. Maro, L. C. Passaro (Eds.), Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online event, December 17th, 2020, volume 2765 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: <https://ceur-ws.org/Vol-2765/paper169.pdf>.
- [23] K. Dubey, Evaluating the fairness of task-adaptive pretraining on unlabeled test data before few-shot text classification, in: D. Hupkes, V. Dankers, K. Batsuren, A. Kazemnejad, C. Christodoulopoulos, M. Giulianelli, R. Cotterell (Eds.), Proceedings of the 2nd GenBench Workshop on Generalisation (Benchmarking) in NLP, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 1–26. URL: <https://aclanthology.org/2024.genbench-1.1/>. doi:10.18653/v1/2024.genbench-1.1.
- [24] N. Auletta, Nicla at DeSegMa-IT: DistilBERT for MGT detection and LightGBM regressor for HMT segmentation., in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [25] H. N. Phu, B. H. Son, D. V. Thin, Stochastic Gradient Descenders at DeSegMa-IT: Instruction-tuned LLM and token classification for MGT detection and boundary localization, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [26] L. Parisi, S. Francia, P. Magnani, UmBERTo: an italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020.
- [27] F. Giobergia, MINDS at DeSegMa-IT: Detecting human-LLM authorship switches via token-level classification, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [28] P. He, J. Gao, W. Chen, DeBERTav3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing, in: The Eleventh International Conference on Learning Representations, 2023. URL: <https://openreview.net/forum?id=sE7-XhLxHA>.
- [29] B. Emi, M. Spero, Technical report on the pangram ai-generated text classifier, 2024. URL: <https://arxiv.org/abs/2402.14873>. arXiv:2402.14873.

A. Generative Models Details

Table 6 reports the four models used to create the machine-generated part of the training set of Sub-task A. Table 7 reports the four models used to create the machine-generated part of the test set of Sub-task A.

A.1. Sub-task A

Model	Hugging Face URL
Qwen3-4B-Instruct	Qwen/Qwen3-4B-Instruct-2507
Gemma-3-4B-it	google/gemma-3-4b-it
Nemo-Instruct	mistralai/Mistral-Nemo-Instruct-2407
Gpt-oss-20B	openai/gpt-oss-20b

Table 6
Training models used, with their relative Hugging Face URLs.

Model	Relative URL
ANITA-NEXT-24B	m-polignano/ANITA-NEXT-24B-Magistral-2506-ITA
Llama-3.3-70B-Instruct	meta-llama/Llama-3.3-70B-Instruct
Minerva-7B	sapienzanlp/Minerva-7B-instruct-v1.0
Gemma-3-27B-it	google/gemma-3-27b-it

Table 7

Models selected for evaluation, with Hugging Face relative URLs.

A.2. Sub-task B

Table 8 reports the nine models used to generate the continuation of human-written news in the train and test sets of Sub-task B.

Model	Relative URL
SmolLM3-3B	HuggingFaceTB/SmolLM3-3B
Qwen3-4B-Instruct	Qwen/Qwen3-4B-Instruct-2507
Gemma-3-27b-it	google/gemma-3-27b-it
ANITA-NEXT-24B	m-polignano/ANITA-NEXT-24B-Magistral-2506-ITA
Llama-3.1-8B-Instruct	meta-llama/Llama-3.1-8B-Instruct
Llama-3.3-70B-Instruct	meta-llama/Llama-3.3-70B-Instruct
Nemo-Instruct	mistralai/Mistral-Nemo-Instruct-2407
Gpt-oss-20b	openai/gpt-oss-20b
Minerva-7B	sapienzanlp/Minerva-7B-instruct-v1.0

Table 8

Generator models used in sub-task B, with Hugging Face relative URLs.

Gradient Descenders at DeSegMa-IT: Leveraging Monolingual Transformer for LLM-Generated Text Detection and Boundary Identification

Tran Phuoc Thanh Nhan^{1,2,*}, Bui Hong Son^{1,2} and Dang Van Thin^{1,2}

¹University of Information Technology, Ho Chi Minh City, Vietnam

²Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

In this paper, we present the Gradient Descenders systems developed for the DeSegMa-IT shared task at EVALITA 2026. Our approach yielded highly competitive results: we achieved the 1st rank in Subtask A with an accuracy of 94.58% and secured the 3rd rank in Subtask B with a Mean Absolute Error (MAE) of 62.66. In the Machine-Generated Text (MGT) detection task, we demonstrated the effectiveness of UmBERTo, a RoBERTa-based model pre-trained on Italian text, validating the advantage of monolingual baselines over multilingual counterparts. In the Boundary Detection task, we approached the problem as a fine-grained token classification task using DeBERTa-base-Italian as the backbone encoder. Looking ahead, we aim to address the identified limitations regarding domain generalization and sequence dependency. Future work will focus on integrating Domain-Adversarial Neural Networks (DANN) to handle domain shifts and exploring more complex classification layers, such as Bi-LSTM, to further refine boundary localization precision.

Keywords

Machine-Generated Text Detection, Boundary Identification, Transformer Models, Italian NLP

1. Introduction

The rapid proliferation of Large Language Models (LLMs), such as GPT-4, Claude, and LLaMa [1], has fundamentally transformed the landscape of digital content creation. While these models offer unprecedented capabilities in generating coherent and contextually relevant text, they also introduce significant challenges regarding information integrity, academic honesty, and the potential for large-scale misinformation. As Machine-generated content becomes increasingly indistinguishable from human-written text, the need for robust, automated detection systems has never been more urgent.

To address these challenges within the Italian landscape, the DeSegMa-IT [2] shared task at EVALITA 2026 [3] has been established. The task focuses on the nuanced differentiation between human and machine authorship and is structured into two distinct but complementary subtasks:

- **Subtask A: MGT Detection in the Wild.** This subtask requires systems to determine whether a given text snippet was authored by a human or generated by an LLM.
- **Subtask B: Human-Machine Text Segmentation.** This subtask proposes human-machine mixed text detection, which provides a text where the first part is human-written and the second part is LLM-generated. The goal is to identify the exact point where the transition occurs.

In this paper, we describe our participation in both of the subtasks of the DeSegMa-IT Shared Task. Our core philosophy involves moving away from generic multilingual models to leverage language-specific adaptations that better capture the morphological richness of Italian. In the context of the DeSegMa-IT shared task, which focuses on Italian, using multilingual models such as XLM-RoBERTa [4] and mDeBERTa [5] may fail to capture the subtle linguistic nuances and morphological richness of

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ 24521241@gm.uit.edu.vn (T. P. T. Nhan); 22521246@gm.uit.edu.vn (B. H. Son); thindv@uit.edu.vn (D. V. Thin)

🆔 0009-0007-2395-3567 (T. P. T. Nhan); 0009-0006-7420-9212 (B. H. Son); 0000-0001-8340-1405 (D. V. Thin)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

the language. Therefore, we propose a language-specific approach leveraging monolingual Italian pre-trained models.

For subtask A, we treat the problem as a sequence classification task. We employ UmBERTo [6], a RoBERTa-based model trained on the massive Italian component of the OSCAR corpus (Open Super-large Crawled ALMAAnaCH coRpus). This choice is motivated by UmBERTo’s superior ability to model Italian syntax and semantics compared to standard multilingual baselines, allowing for more accurate discrimination between human and machine stylistic patterns.

For subtask B, we approach the boundary detection challenge as a fine-grained Token classification problem. We utilize DeBERTa-base-Italian [7] as our backbone encoder. Technically, this model is built upon the mDeBERTa-v3-base [5] architecture. To focus on the Italian language, the backbone has been adapted to Italian by replacing the original embedding layer [8] and undergoing continued pre-training on Italian data. By attaching a linear classification head on top of this specialized backbone, our system predicts a binary label for every token, enabling it to pinpoint the precise transition index where the authorship shifts from human to machine with high granularity.

The effectiveness of our proposed approaches is validated through extensive experiments on the official datasets provided by the DeSegMa-IT shared task. Our systems demonstrated strong performance in both subtasks:

- **For Subtask A:** Our UmBERTo-based approach achieved state-of-the-art results, securing the 1st rank on the leaderboard with an accuracy of 0.945755.
- **For Subtask B:** Our token classification strategy using DeBERTa backbone yielded competitive results, achieving a Mean Absolute Error (MAE) of 62.66, placing us 3rd on the leaderboard.

This paper is structured as follows: In Section 2, we review related work on LLM-generated text detection and boundary detection in human-machine mixed texts. We also highlight some studies and their limitations, then position our work in this context. In Section 3, we provide a detailed overview of our proposed systems for both subtasks, including model architectures and training strategies. In Section 4, we describe the datasets used, evaluation metrics for each subtask, and our experimental setup. In Section 5, we present and discuss the results obtained by our systems on the official test sets. In Section 6, we analyze the strengths and limitations of our approaches and suggest potential avenues for future research. Finally, we conclude the paper in Section 7 and summarize our findings.

2. Related Work

2.1. Detecting LLM-Generated Text

The detection of LLM-generated text is often framed as a binary classification problem, where the objective is to train a classifier to distinguish between human-written text and LLM-generated text [9]. One traditional approach involves employing statistical techniques to identify generation artifacts. Such approaches utilize features including perplexity, entropy, Zipfian distribution of words, and n-gram frequency analysis. Feature-based methodologies typically operate by constructing high-dimensional vectors representing various stylistic and statistical properties of the text. Once extracted, these feature vectors serve as inputs for traditional machine learning classification algorithms, such as a support-vector machine (SVM), random forest (RF). For instance, Gehrmann et al. [10] developed a system that combines logistic regression with GLTR, a tool employing baseline statistical methods to detect generation artifacts across common sampling schemes.

Another prominent line of research involves leveraging Neural Language Models (NLMs) for text classification. Several works have fine-tuned large bidirectional language models, such as BERT and RoBERTa [11], on datasets containing both human-written and machine-generated text. These models utilize self-attention mechanisms to capture long-range dependencies and semantic coherence, which are often the hallmarks of neural generation. For example, Solaiman et al. [12] demonstrated that a fine-tuned RoBERTa model could achieve detection accuracy upward of 90% on GPT-2 generated text, significantly outperforming statistical baselines like GLTR. Adopting a generative approach, Chen et

al. [13] utilized the Text-to-Text Transfer Transformer (T5) classifier on a dataset named OpenLLMText, and then used these classifiers to predict the conditional probability of the next word, thereby classifying multiple text sources. By framing MGT detection as a text-to-text generation task, they utilized the model’s perplexity and next-token probabilities to classify texts from diverse sources, demonstrating superior generalization compared to traditional classification heads. Zellers et al. [14] introduced ‘Grover’, a model designed to defend against neural fake news. They empirically showed that the most effective detector for a specific generator is often the generator itself, fine-tuned as a discriminator. Wang et al. [15] proposed SeqX-GPT, a method that utilizes log probability lists from white-box LLMs as features for sentence-level AIGT detection.

Furthermore, while multilingual models like mBERT [16] and XLM-RoBERTa [4] provide strong baselines, previous studies on the ‘curse of multilinguality’ [4] indicate that allocating model capacity across too many languages can dilute performance on individual ones. Empirical evidence was provided by Martin et al. [17] in the context of French NLP tasks, where they demonstrated that CamemBERT, a RoBERTa-based model trained exclusively on French text, outperformed multilingual counterparts by a significant margin.

2.2. Boundary Detection for Human-Machine Mixed Text

Beyond numerous studies focused on machine-generated text detection, a less-explored yet significant challenge lies in identifying precise boundaries within machine-human mixed texts. These texts consist of two segments: the first part is human-written, and the second part is generated by LLMs. Dugan et al. [18] conducted a comprehensive study to understand human ability to discern boundaries between human-written and machine-generated text. Their findings revealed significant variations in annotator proficiency and highlighted the impact of various factors on detection accuracy, such as model size, decoding strategy, fine-tuning,... etc. Zeng et al. [19] were among the first to formally define the task of identifying transition points between human-written and LLM-generated content. They proposed a two-step approach aimed to separate human and machine representations. First, they separate the LLM-generated text from human-written text during the encoder training process. Second, they calculated the distances between adjacent text prototypes, positing that the transition boundary is located between the two prototypes exhibiting the furthest distance from each other.

While these pioneering studies have proposed effective methodologies for boundary detection in hybrid texts, they were typically limited to the sentence level rather than at the token level. As a consequence, they may lead to poor granularity in realistic writing scenarios, where an LLM might complete a sentence started by a user. Addressing this gap, we approach boundary detection as a sequence labeling problem, enabling precise identification of transitions at the token level.

3. System Overview

Our systems for both subtasks are built upon pre-trained Italian transformer-based models. We utilize these models as backbone encoders and attach task-specific heads to adapt them to the respective tasks.

In Subtask A, we employ UmBERTo [6] as our backbone encoder for binary classification. We append a linear classification head on top of the [CLS] token representation to predict whether the input text is human-written or LLM-generated. See Section 3.1 for more details.

In Subtask B, the target is to identify the exact token index where the transition from human-written to machine-generated text occurs. As mentioned in Section 2.2, previous studies typically approached this problem at the sentence level, which may not provide sufficient granularity for practical applications. Therefore, we approach this task as a sequence labeling problem, employing DeBERTa-base-Italian [7] as our backbone encoder and attaching a token-level classification head to predict binary labels for each token. See Section 3.2 for more details.

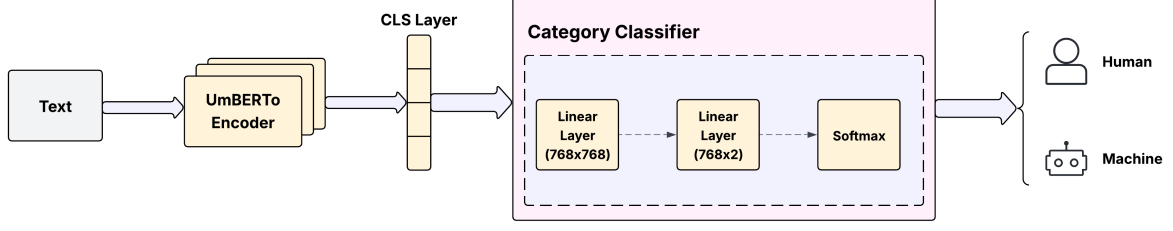


Figure 1: System overview for Subtask A

3.1. Subtask A: Binary Classification with UmBERTo

The overview of our system is illustrated in Figure 1. Our proposed system for Subtask A consists of two main components: a feature extractor layer (such as UmBERTo) that acquires text representation and a category classifier that predicts whether the input text is human-written or LLM-generated.

The feature extractor and the category classifier constitute a feedforward neural network architecture. Specifically, the feature extractor $G_f(\cdot)$ initially receives the text input, which consists of n tokens $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, and then transforms it into high-dimensional contextualized embeddings. We utilize the output vector corresponding to the CLS token as the aggregate semantic summary of the entire sequence, denoted as $G_f(\mathbf{x})$. Subsequently, this semantic representation $G_f(\mathbf{x})$ is passed to the category classifier $G_y(\cdot)$, yielding $G_y(G_f(\mathbf{x}))$, representing the predicted probability of two labels through the Softmax function. Finally, the predicted label $\hat{y}$ is determined by selecting the class with the highest probability.

In the category classifier, we employ a two-layer Multi-Layer Perceptron (MLP) with a Tanh activation function between the layers. We then apply a Dropout operation to mitigate overfitting before passing the features through a second linear layer to map them to the binary label space.

To train the model, we employ the Cross-Entropy loss function, which minimizes the discrepancy between the predicted probabilities and the true labels. The loss function is defined as follows:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \log(\hat{y}_{ij}) \quad (1)$$

where N is the number of samples in the batch, C is the number of classes (2 in this case), y_{ij} is the true label (one-hot encoded) for class j of sample i , and $\hat{y}_{ij}$ is the predicted probability for class j of sample i (obtained through the Softmax function).

3.2. Subtask B: Fine-Grained Boundary Detection via Token Classification

In subtask B, the objective is to identify the exact token index where the transition from human-written text to machine-generated text occurs. As mentioned in Section 2.2, we approach this task as a sequence labeling problem at the token level.

One of the main challenges in this subtask is aligning the ground-truth boundary index with the tokenized sequence produced by the tokenizer of the pre-trained model. To address this, we utilize the tokenizer’s offset mapping feature, which provides the start and end character positions of each token in the original text. For each input sentence, the tokenizer generates a mapping (s_i, e_i) for each token t_i , where s_i and e_i denote the start and end character indices of the token in the original text, respectively. Then each token t_i is assigned a binary label y_i based on the ground-truth boundary index as follows:

$$y_i = \begin{cases} 0(\text{Human}) & \text{if } s_i < C_{trans} \\ 1(\text{Machine}) & \text{else} \end{cases} \quad (2)$$

where C_{trans} is the ground-truth character index of the transition point.

Table 1

Statistics of the DeSegMa-IT datasets for Subtask A and Subtask B

Subtask	Category	Train	Dev	Test
Subtask A	Human	16,569	–	–
	Machine	16,569	–	–
	Total	33,138	5,059	15,135
Subtask B	Total	19,945	4,579	23,211

We employ DeBERTa-base-Italian [7] as our backbone encoder for feature extraction. On top of the feature extractor, we attach a linear classification head that predicts binary labels for each token in the input sequence. Technically, the input sequence is initially encoded into hidden state $H = \{h_1, \dots, h_n\}$. These states are then passed through the classification head, which consists of a single linear layer followed by a Softmax function, yielding predicted probabilities for each token:

$$P(\hat{y}_i|x_i) = \text{softmax}(W \cdot h_i + b) \quad (3)$$

Between the backbone encoder and the classification head, we apply a Dropout layer to mitigate overfitting.

4. Experimental setup

4.1. Datasets and Evaluation Metrics

Datasets We utilized the official datasets provided by organizers of the DeSegMa-IT [2] shared task. The corpus is exclusively in Italian for both subtasks.

- **For subtask A:** The dataset consists of over 33000 unique text samples, which are labeled as either human-written (0) or LLM-generated (1). The dataset is split into training, validation, and test sets. The texts vary in length from short paragraphs to longer articles, covering a diverse range of topics and styles.
- **For subtask B:** The dataset contains nearly 20000 text samples. Each document begins with a human-authored segment and transitions into machine-generated text at a specific, unmarked boundary. The objective is to pinpoint the exact token index of this transition.

See Table 1 for more details about the datasets of both Subtask A and Subtask B.

Evaluation Metrics The evaluation metric used in subtask A is accuracy, which is defined as the ratio of correctly predicted samples to the total number of samples. For subtask B, the evaluation metric is Mean Absolute Error (MAE), which measures the average absolute difference between the predicted boundary index and the actual boundary index. The performance is better when the MAE value is lower.

4.2. Training

Subtask A In subtask A, we divided the training into two phases: (1) fine-tuning the UmBERTo [6] model on the training set and (2) hyperparameter tuning on the validation set with the ratio of 80:20. For input processing in Subtask A, given the maximum context window of 512 tokens inherent to UmBERTo, we employed a standard truncation strategy, retaining only the initial 512 tokens of each document. The entire model, including both the UmBERTo backbone and the classification head, was fine-tuned end-to-end using the AdamW [20] optimizer coupled with a linear learning rate scheduler. The learning rate scheduler was configured with a fixed warm-up period of 100 steps. The model

Table 2

Official Leaderboard Results for Subtask A.

Team	Accuracy	Rank
Kenji Endo	0.9426	2
UniTor	0.9288	3
Niclas	0.9243	4
Stochastic G.D.	0.9216	5
Gradient Descenders	0.9458	1

Table 3

Ablation Study: Monolingual vs. Multilingual.

Model	Accuracy	Macro-F1	Micro-F1
mBERT (Base)	0.9393	0.9392	0.9393
XLM-RoBERTa (Base)	0.9397	0.9396	0.9397
mDeBERTa (Base)	0.9248	0.9245	0.9248
UmBERTo (Ours)	0.9458	0.9376	0.9378

is optimized by minimizing the Cross-Entropy loss function. To mitigate overfitting, we applied an Early Stopping strategy based on the validation loss, with a patience of 5 epochs. Furthermore, we incorporated a Dropout layer within the classification head and weight decay during optimization. For reproducibility, a comprehensive list of all hyperparameters, search spaces, and hardware specifications is provided in Appendix A.1

Subtask B In subtask B, we followed a similar training procedure as in Subtask A. We divided the dataset into training and validation sets with an 80:20 ratio. For input processing in Subtask B, we also truncated the input sequences to a maximum length of 512 tokens and assigned the special tokens ([CLS], [SEP]) with the label -100 to exclude them from loss computation. The entire model, including the DeBERTa-base-Italian [7] model and classification layer, was trained by minimizing the Cross-Entropy loss function using the AdamW [20] optimizer with weight decay. We also maintained the linear learning rate scheduler with a fixed warm-up period of 100 steps. During the evaluation, we observed that the token-level predictions can be noisy, resulting in unstable boundary detection and spurious label flipping. To address this issue, we implemented a consecutive consistency constraint during the inference phase. Specifically, a token at index k was identified as the transition point only if the model predicted the label 1 (Machine) for at least m consecutive tokens starting from index k . This approach effectively filters out isolated misclassifications and enhances the robustness of boundary detection. For reproducibility, more details about the hyperparameters and hardware specifications are provided in Appendix A.2.

5. Results

In this section, we present and discuss the results obtained by our proposed systems on both subtasks of the DeSegMa-IT [2] shared task. In subtask A, our system achieved the highest accuracy on the official test set, securing the 1st position on the leaderboard. In subtask B, our approach demonstrated competitive performance, achieving the 3rd rank on the leaderboard.

5.1. Subtask A: MGT Detection in the Wild

Table 2 presents the official leaderboard for Subtask A. Our proposed system, submitted under the team name *Gradient Descenders*, achieved the top position with an accuracy of 0.945755. As observed, the competition was intense, particularly between our system and the runner-up (*Kenji Endo*), with a margin

Table 4

Official Leaderboard Results for Subtask B.

Team	MAE	Rank
Stochastic G.D.	52.54	1
MINDS	56.53	2
UniTor	81.6	4
Nicla	102.04	5
Gradient Descenders	62.66	3

Table 5

Ablation Study: Backbone Comparison.

Model	MAE
UmBERTo (Base)	65.22
mDeBERTa-v3 (Base)	63.34
DeBERTa-IT (Ours)	62.66

of approximately 0.3%. However, our model demonstrated a significant lead over other competitors, outperforming the third-ranked team (*UniTor*) by over 1.7 percentage points.

To validate the effectiveness of our design choices, Table 3 compares our monolingual backbone against various baselines. The result empirically supports our hypothesis that leveraging a monolingual Italian pre-trained model, specifically UmBERTo [6], provides a substantial advantage in capturing the linguistic nuances necessary for effective discrimination between human and machine-generated text. Notably, UmBERTo outperforms the multilingual XLM-RoBERTa, which shares a similar architecture, by a margin of 0.64%, highlighting the benefit of language-specific pre-training. Moreover, it is particularly instructive to compare our system with *Stochastic Gradient Descenders*, a team from our same research group that employed a fundamentally different approach. While our system relies on an encoder-only architecture with full fine-tuning, the *Stochastic Gradient Descenders* team employed a generative Large Language Model (LLM) (Qwen2.5–0.5B model [21]) adapted via Low-Rank Adaptation (LoRA) [22]. Although Qwen2.5–0.5B is a highly effective multilingual model, our proposed system outperforms it by 2.42%. This result indicates that for the specific task of MGT detection in Italian, a well-tuned encoder-based model can outperform a generative LLM approach.

5.2. Subtask B: Human-Machine Text Segmentation

Table 4 shows the official leaderboard for Subtask B. Our system, *Gradient Descenders*, secured the 3rd position with a Mean Absolute Error (MAE) of 62.66. The system exhibited strong performance, establishing a significant gap over lower-ranked teams; specifically, we outperformed *Nicla* and *UniTor* by margins of 39.38 and 18.94 points, respectively.

Table 5 presents an ablation study comparing different backbone architectures for the boundary detection task. The empirical experiments have presented the effectiveness of both the DeBERTa architecture and monolingual pre-training. DeBERTa-IT surpasses its multilingual counterparts, mDeBERTa-V3 (63.34), restating the “monolingual advantage” observed in Subtask A.

It is noteworthy that our proposed system solely relied on a simple token classification layer, without incorporating more sophisticated techniques such as Conditional Random Fields (CRF) [23] or Bi-LSTM [24] layers. While computational budget constraints limited our exploration of larger backbones or more complex architectures, this result confirms that a well-tuned baseline with robust post-processing can achieve top-tier performance. Consequently, we believe there is ample room for improvement by scaling up the model size or integrating sequence modeling layers in future work.

6. Discussion

6.1. Subtask A Analysis

Our performance in Subtask A demonstrates the effectiveness of using a monolingual Italian pre-trained model, UmBERTo [6], for the task of MGT detection. These results align with prior findings on the ‘curse of multilinguality’ [4], which suggest that language-specific models can better capture linguistic nuances compared to multilingual counterparts. Although the performance gap is modest and comparable in terms of F1-score, we hypothesize that scaling up the volume and diversity of the Italian pre-training

data could significantly amplify this advantage, allowing the monolingual model to fully leverage its specialized capacity.

However, a notable challenge observed is the system’s sensitivity to domain shift, where the distribution of the test data diverges from the training set. This is a challenge commonly faced in real-world applications, where the distribution of test data may differ significantly from the training data. To mitigate this limitation in future work, we propose exploring domain adaptation techniques. Guo et al. [25] addressed this issue by employing DANN (Domain-Adversarial Neural Networks) [26] to learn domain-invariant features, thereby enhancing the model’s robustness to domain shifts. By employing a gradient reversal layer, such architectures can effectively align feature distributions across domains, thereby enhancing robustness.

Furthermore, augmenting the training data with synthetic examples generated by various LLMs could further enhance the model’s generalization capabilities. The diversity in generation styles and artifacts in different LLMs can provide a richer training signal, enabling the model to generalize better to unseen data.

6.2. Subtask B Analysis

As discussed in Section 5, our approach for Subtask B, which treats boundary detection as a token classification problem, yielded competitive results despite its simplicity. The performance shows the strong ability of DeBERTa-base-Italian [7] in capturing token-level nuances necessary for identifying transition points between human and machine-generated text. Furthermore, this also validates our hypothesis that token-level classification can provide finer granularity compared to sentence-level approaches discussed in Section 2.2.

Despite these strengths, we acknowledge several limitations. First, our backbone model (DeBERTa-base-Italian) was fine-tuned on a limited corpus, which may limit its generalization capabilities. Second, while the DeBERTa encoder captures input context effectively, the simple linear classification head treats each token’s prediction independently. It lacks the mechanism to explicitly model output label dependencies, potentially leading to misclassifications around the transition boundary. Guo et al. [25] have experimented by using Bi-LSTM [24] layers on top of pre-trained models for similar token-level tasks, providing a significant performance boost rather than using only linear layers. Finally, the fixed truncation at 512 tokens is a bottleneck for long-document analysis. Future iterations will investigate models with extended context windows (e.g., Longformer) or sliding window inference strategies to address boundaries situated deep within the text.

7. Conclusion

In this paper, we presented the Gradient Descenders systems developed for the DeSegMa-IT [2] shared task at EVALITA 2026 [3]. Our approach yielded highly competitive results: we achieved the 1st rank in Subtask A with an accuracy of 94.58% and secured the 3rd rank in Subtask B with a Mean Absolute Error (MAE) of 62.66. These findings empirically demonstrate the effectiveness of leveraging monolingual Italian pre-trained models, specifically UmBERTo [6] and DeBERTa-base-Italian [7], for MGT detection and boundary detection tasks, respectively. We also discussed the strengths and limitations of our approaches, providing insights into potential avenues for future research. Looking ahead, we plan to enhance model robustness against domain shifts using Domain-Adversarial Neural Networks (DANN) [26] and to refine boundary precision by integrating structured prediction layers, such as Bi-LSTM [24], into our pipeline.

Declaration on Generative AI

Generative AI tools were employed only to enhance linguistic clarity. All data, analyses, and conclusions were conceived, executed, and validated by the authors. The authors retain full accountability for every

aspect of the manuscript’s content.

References

- [1] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. URL: <https://arxiv.org/abs/2302.13971>. arXiv:2302.13971.
- [2] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [4] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020. URL: <https://arxiv.org/abs/1911.02116>. arXiv:1911.02116.
- [5] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2021. arXiv:2111.09543.
- [6] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020.
- [7] Osiria, Deberta-base-italian: Italian deberta model, <https://huggingface.co/osiria/deberta-base-italian>, 2023. Pre-trained Italian language model based on DeBERTa architecture.
- [8] A. Abdaoui, C. Pradel, G. Sigel, Load what you need: Smaller versions of multilingual bert, 2020. URL: <https://arxiv.org/abs/2010.05609>. arXiv:2010.05609.
- [9] H.-Q. Nguyen-Son, N.-D. T. Tieu, H. H. Nguyen, J. Yamagishi, I. E. Zen, Identifying computer-generated text using statistical analysis, in: 2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017, pp. 1504–1511. doi:10.1109/APSIPA.2017.8282270.
- [10] S. Gehrmann, H. Strobelt, A. M. Rush, Gltr: Statistical detection and visualization of generated text, 2019. URL: <https://arxiv.org/abs/1906.04043>. arXiv:1906.04043.
- [11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: <https://arxiv.org/abs/1907.11692>. arXiv:1907.11692.
- [12] I. Solaiman, M. Brundage, J. Clark, A. Asbell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, J. Wang, Release strategies and the social impacts of language models, 2019. URL: <https://arxiv.org/abs/1908.09203>. arXiv:1908.09203.
- [13] Y. Chen, H. Kang, V. Zhai, L. Li, R. Singh, B. Raj, Token prediction as implicit classification to identify LLM-generated text, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 13112–13120. URL: <https://aclanthology.org/2023.emnlp-main.810/>. doi:10.18653/v1/2023.emnlp-main.810.
- [14] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against neural fake news, 2020. URL: <https://arxiv.org/abs/1905.12616>. arXiv:1905.12616.
- [15] P. Wang, L. Li, K. Ren, B. Jiang, D. Zhang, X. Qiu, Seqxgpt: Sentence-level ai-generated text detection, 2023. URL: <https://arxiv.org/abs/2310.08903>. arXiv:2310.08903.
- [16] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers

- for language understanding, CoRR abs/1810.04805 (2018). URL: <http://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [17] L. Martin, B. Muller, P. J. Ortiz Suarez, Y. Dupont, L. Romary, E. de la Clergerie, D. Seddah, B. Sagot, Camembert: a tasty french language model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 7203–7219. URL: <http://dx.doi.org/10.18653/v1/2020.acl-main.645>. doi:10.18653/v1/2020.acl-main.645.
 - [18] L. Dugan, D. Ippolito, A. Kirubarajan, S. Shi, C. Callison-Burch, Real or fake text?: Investigating human ability to detect boundaries between human-written and machine-generated text, 2022. URL: <https://arxiv.org/abs/2212.12672>. arXiv:2212.12672.
 - [19] Z. Zeng, L. Sha, Y. Li, K. Yang, D. Gašević, G. Chen, Towards automatic boundary detection for human-ai collaborative hybrid essay in education, 2023. URL: <https://arxiv.org/abs/2307.12267>. arXiv:2307.12267.
 - [20] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: <https://arxiv.org/abs/1711.05101>. arXiv:1711.05101.
 - [21] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Yang, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, X. Liu, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, Z. Guo, Z. Fan, Qwen2 technical report, 2024. URL: <https://arxiv.org/abs/2407.10671>. arXiv:2407.10671.
 - [22] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, 2021. URL: <https://arxiv.org/abs/2106.09685>. arXiv:2106.09685.
 - [23] C. Sutton, A. McCallum, An introduction to conditional random fields, 2010. URL: <https://arxiv.org/abs/1011.4088>. arXiv:1011.4088.
 - [24] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: K. Erk, N. A. Smith (Eds.), Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 207–212. URL: <https://aclanthology.org/P16-2034/>. doi:10.18653/v1/P16-2034.
 - [25] Z. Guo, K. Jiao, X. Yao, Y. Wan, H. Li, B. Xu, L. Zhang, Q. Wang, Y. Zhang, Z. Mao, USTC-BUPT at SemEval-2024 task 8: Enhancing machine-generated text detection via domain adversarial neural networks and LLM embeddings, in: A. K. Ojha, A. S. Doğruöz, H. Tayyar Madabushi, G. Da San Martino, S. Rosenthal, A. Rosá (Eds.), Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 1511–1522. URL: <https://aclanthology.org/2024.semeval-1.217/>. doi:10.18653/v1/2024.semeval-1.217.
 - [26] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, V. Lempit-sky, Domain-adversarial training of neural networks, 2016. URL: <https://arxiv.org/abs/1505.07818>. arXiv:1505.07818.

A. Detailed Experimental Setup

A.1. Subtask A: Binary Classification with UmbERTO

We utilized UmbERTO (Cased-v1) [6] as our backbone encoder for Subtask A, coupled with the two-layer MLP classification head described previously. The model was fine-tuned using the AdamW [20] optimizer with a weight decay of $1e-2$ to regularize the weights. We employed a linear learning rate scheduler with a warm-up period of 100 steps. After the warm-up phase, the learning rate decayed linearly to zero.

Regarding hyperparameters, the peak learning rate was set to $2e-5$ with a batch size of 8. The input

sequences were truncated to a maximum length of 512 tokens. As mentioned in Section 4, we employed an Early Stopping strategy based on the validation loss, with a patience of 5 epochs. The Dropout rate in the classification head was set to 0.2. We conducted training for 10 epochs on NVIDIA Tesla T4 GPUs.

A.2. Subtask B: Human-Machine Text Segmentation

We utilized DeBERTa-base-Italian [7] as our backbone encoder for Subtask B, along with the token-level classification head described previously. The model was trained using the AdamW [20] optimizer with a weight decay of $1e-2$. We employed a linear learning rate scheduler with a warm-up period of 100 steps, similar to Subtask A.

Regarding hyperparameters, the learning rate was set at $1e-5$, and the batch size was 8. Input sequences were truncated to a maximum length of 512 tokens, with special tokens ([CLS], [SEP]) assigned a label of -100 to exclude them from loss computation. A dropout with a rate of 0.3 was applied before the classification head to mitigate overfitting. For the consecutive consistency constraint during inference, we set the parameter m to 2, meaning that a token is classified as the transition point only if at least 2 consecutive tokens are predicted as machine-generated starting from that token. The training phase was conducted for 6 epochs on NVIDIA Tesla T4 GPUs.

MINDS at DeSegMa-IT: Detecting Human-LLM Authorship Switches via Token-Level Classification

Flavio Giobergia

Politecnico di Torino. Turin, Italy

Abstract

We address the DeSegMa-IT shared task at EVALITA, which aims to identify and segment mixed-authorship Italian text, where generation switches from a human prompt to a large language model continuation. We address the problem as a token-level sequence labeling, where each token is labelled as either human- or machine-generated. To this end, we use an encoder-only transformer, fine-tuned for token-level binary token classification (human vs. LLM), with a post-hoc threshold tuning step to improve boundary localization. Our system achieves competitive performance, ranking second in the official evaluation. In this report, we summarize the proposed solution, and explore additional options considered during the challenge.

Keywords

machine-generated text detection, authorship segmentation, large language models

1. Introduction

In recent years, rapid progress in generative artificial intelligence has led to the widespread adoption of large language models (LLMs) capable of producing text that is often difficult to distinguish from human-written content. Such models are now routinely employed in applications ranging from creative writing [1] and customer support tasks [2], to the drafting [3] and reviewing [4] of scientific papers. However, alongside these benefits, the usage of LLMs raises increasing concerns related to transparency, accountability, factual validity and misuse. Indeed, language models can be used to generate misinformation, fake news, conspiracy theories and hate content at a previously impossible rate. Such contents already flood places such as news outlets or social media. In response to these risks, regulatory frameworks – most notably the European AI Act – are increasingly emphasizing the need for mechanisms that automatically recognize machine-generated contents reliably.

Within this context, the DeSegMa-IT [5] shared task at EVALITA 2026 [6] addresses the challenge of machine-generated text detection for the Italian language, with a particular focus on “non-ideal” settings. Rather than operating under fully controlled conditions, the task is designed to evaluate the robustness of detection systems in scenarios where prior knowledge of the generation process may be unavailable. Among the proposed subtasks, Task B focuses on a segmentation problem: given a single text composed of an initial human-written prefix followed by an LLM-generated continuation, the goal is to identify the precise point at which authorship changes.

Unlike document-level detection, SubTask B explicitly emphasizes *localization*: the solutions evaluated using the mean absolute error between the predicted and gold switch *character position*. The approach of detecting an event at finer-granularities is commonly adopted in LLM-generated contents: other works in literature focused, for instance, on the detection of token-level toxic content, or the detection of hallucinations at the character level [7]. In this work, we address SubTask B by formulating it as a token-level problem, using an encoder-only transformer model (based on BERT [8]), fine-tuned on a token-level classification task. The model assigns authorship labels to individual tokens, from which the switch position is derived, allowing for fine-grained modeling of the human-machine boundary.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ flavio.giobergia@polito.it (F. Giobergia)

🆔 0000-0001-8806-7979 (F. Giobergia)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

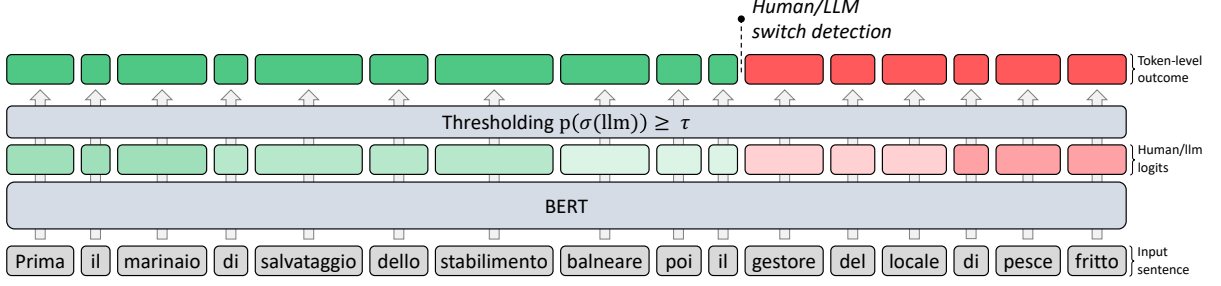


Figure 1: Intuition behind the proposed solution: tokenized sentences are annotated by a suitably fine-tuned token-level classifier and later converted to either human- or machine-generated.

2. Methodology

The proposed methodology is based on the idea of letting a model distinguish between human and LLM-generated tokens, at a token level. In other words, we build a transformer-based model capable of annotating each token of a sentence as either human- or LLM-generated. The transformer architecture is particularly well-suited for this kind of tasks, where the model can only assign a valid token-level prediction by considering the surrounding context.

We summarize the intuition in Figure 1, which shows the main steps of the proposed solution: a first token-level prediction, the following conversion to a binary outcome (human/LLM), and the final identification of the switch point (i.e., the first transition from human-predicted to machine-predicted token).

2.1. Task formulation

Let a text be tokenized into a sequence $(t_1, t_2, \dots, t_n)$. The goal is to predict the switch point where the text transitions from human-written to LLM-generated. We frame the problem as binary token classification by assigning each token t_i a label $y_i \in \{0, 1\}$, where 0 denotes *human* and 1 denotes *LLM*. The gold labeling is derived from the annotated switch: tokens before the switch are labeled 0 and tokens at/after the switch are labeled 1.

A transformer encoder (BERT-style) produces contextual token representations $\mathbf{h}_i$, followed by a linear classification head yielding logits $\mathbf{z}_i \in \mathbb{R}^2$ and probabilities $\mathbf{p}_i = \text{softmax}(\mathbf{z}_i)$.

2.2. Switch prediction

At inference time, we obtain token-level predictions $\hat{y}_i = \mathbf{1}(\mathbf{p}_{i,1} \geq \tau)$. In other words, we assign a class based on whether the probability for LLM-generated is above a certain threshold τ (hyperparameter). We then estimate the switch token index as the earliest position predicted as LLM:

$$\hat{k} = \min\{i \mid \hat{y}_i = 1\},$$

with a fallback to $\hat{k} = n$ if no token is predicted as LLM. The final system output is the corresponding *character position* in the original text, obtained by mapping the predicted switch token back to the position of the token’s starting character.

3. Experimental Results

In this section we report the main results obtained for the challenge. Specifically, we report three aspects of interest: the main results obtained during the challenge (Section 3.1), some additional experiments to identify the most suitable backbone model (Section 3.2) and, finally, a sensitivity analysis to define the hyperparameter τ (Section 3.3). We note that, for experiments that were not submitted during the competition, we report the results based on a separate test set that has been made available throughout the challenge.

Table 1

Official results for the DeSegMa-IT challenge (SubTask B). The best solution is shown in **bold**, the second best solution is underlined.

Position	Participant	MAE ↓
1	Stochastic Gradient Descenders	52.54
2	<i>MINDS (ours)</i>	<u>56.53</u>
3	Gradient Descenders	62.66
4	UniTor	81.6
-	<i>Regression baseline</i>	90.54
5	Nicla	102.04
-	<i>Naive prediction (avg)</i>	103.53

Table 2

Backbone comparison under the same token-classification framework. The best solution is shown in **bold**, the second best solution is underlined.

Architecture	Size (M params)	Pre-training language	MAE ↓
BERT	110	Italian	56.53
RoBERTa	125	Italian	<u>56.95</u>
RoBERTa	355	English	57.95
RoBERTa	125	English	58.99
BERT	340	English	76.92

3.1. Main results

Table 1 reports the official leaderboard results. Our system achieves second place overall, indicating that the proposed token-classification formulation is effective for localizing human-to-LLM transitions in Italian text. We additionally introduce (i) a simple model that addresses the task directly as a regression problem, i.e. using the output for the CLS token, after proper fine-tuning, to predict the switch point, and (ii) the performance for the naive solution that predicts the average starting point, to provide a baseline result.

3.2. Backbone definition

To understand the impact of the underlying encoder, we also compare different backbone models under the same training setup. Table 2 summarizes results across an Italian-trained BERT-style model [9] and RoBERTa-based [10] alternatives (including an Italian-adapted variant [11]). All trainings have been performed for a total of 10 epochs (batch size of 8). Further training did not provide additional generalization benefits, neither to the smaller, nor to the larger models. Overall, the results show that the Italian-tuned BERT model outperforms other models, even larger ones. This appears to indicate that, for this problem, the pre-training is more beneficial than an increased model size. Based on these conclusions, the Italian version of BERT¹ has been used as the backbone for the proposed method.

3.3. Sensitivity analysis

On top of the hyperparameters already defined for the training phase, one additional value of interest is the threshold τ applied to the model’s predicted probabilities, to assign a hard label to each token. This hyperparameter can be tuned *after* the training process, using a separate validation set and studying the change in performance on the validation set, as a function of τ . We report this analysis in Figure 2. The result highlights how the model achieves good performance for small values of τ – indicating that the model is not well-calibrated. There is a limited plateau of stable performance. As the threshold

¹dbmdz/bert-base-italian-cased on HuggingFace

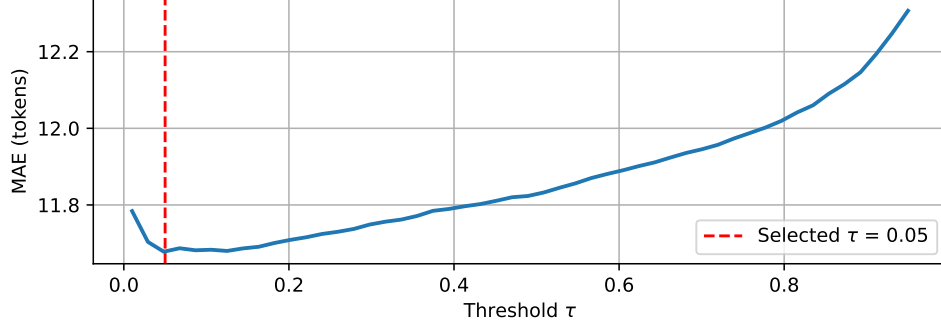


Figure 2: Performance of the model, as a function of τ . The performance is reported in terms of Mean Absolute Error computed at the token level, for consistency with the token-level approach of the proposed solution.

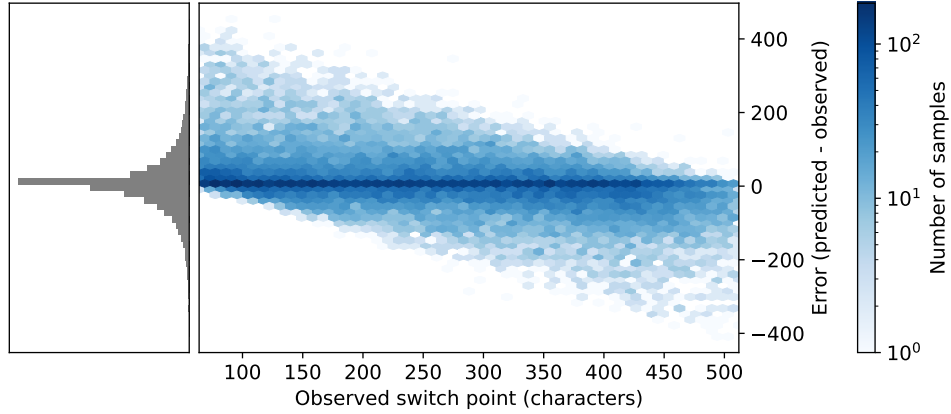


Figure 3: Representation of the residual error, as a function of the observed switch point. The side histogram shows the shape of the distribution of the errors, qualitatively representing a symmetric distribution.

increases, we observe a drop in performance (i.e., an increase in MAE): because of these empirical observations, we set the threshold to $\tau = 0.05$.

3.4. Error analysis

We additionally look into the distribution of residual errors, as shown in Figure 3. The plot shows the residual error of the model (computed as the difference between predicted and observed values), as a function of the observed switch point. The high-density region of space centered around 0 indicates that, for a majority of samples, the model behaves consistently across different switch positions. As expected, the error for “early” switch points is mostly positive (as the model can only be mistaken when predicting late switch points). The same applies for “late” switch points and reversed considerations (i.e., the mistakes are related to predicting an early switch point). We observe that the distribution of errors is symmetric: this intuitively indicates that the model does not overly predict early (or late) switches. Finally, we note that the mean (signed) error approximately 7.95: in other words, since the error is not zero-centered, the additional offset could be subtracted (e.g., if computed on an i.i.d. validation set) to compensate this bias.

4. Conclusions

We presented a token-level approach for the DeSegMa-IT task, casting authorship transition detection as binary sequence labeling with a BERT-style encoder. Our system achieves competitive performance, ranking second in the shared task evaluation. Future work could explore alternative boundary-aware objectives, improved token-to-character alignment strategies, and extensions to other languages or

mixed-generation settings. We attempted additional approaches, e.g. an alternative version of the proposed approach, with weighted importance assigned to the tokens based on their distance from the switch point. However, this alternative achieved comparable performance to the non-weighted version, and we did not report those results for simplicity. Other options, e.g. the direct prediction of the switch point as a regression task, were not found to be promising.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (GPT 5.2) in order to: Drafting content. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] C. Gómez-Rodríguez, P. Williams, A confederacy of models: A comprehensive evaluation of llms on creative writing, arXiv preprint arXiv:2310.08433 (2023).
- [2] V. Scotti, M. J. Carman, Llm support for real-time technical assistance, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2024, pp. 388–393.
- [3] W. Liang, Y. Zhang, Z. Wu, H. Lepp, W. Ji, X. Zhao, H. Cao, S. Liu, S. He, Z. Huang, et al., Mapping the increasing use of llms in scientific papers, arXiv preprint arXiv:2404.01268 (2024).
- [4] F. Giobergia, A. Koudounas, E. Baralis, Large language models-aided literature reviews: A study on few-shot relevance classification, in: 2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT), IEEE, 2024, pp. 1–5.
- [5] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, 2026.
- [6] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [7] C. Savelli, A. Koudounas, F. Giobergia, MALTO at SemEval-2025 task 3: Detecting hallucinations in LLMs via uncertainty quantification and larger model validation, in: S. Rosenthal, A. Rosá, D. Ghosh, M. Zampieri (Eds.), Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 1318–1324. URL: <https://aclanthology.org/2025.semeval-1.175/>.
- [8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.
- [9] B. Staatsbibliothek, S. Schweter, bert-base-italian-cased (revision 843e404), 2025. URL: <https://huggingface.co/dbmdz/bert-base-italian-cased>. doi:10.57967/hf/5850.
- [10] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[11] roberta-base-italian, 2025. URL: <https://huggingface.co/osiria/roberta-base-italian>.

Nicla at DeSegMa-IT: DistilBERT for MGT Detection and LightGBM Regressor for HMT Segmentation

Nicla Auletta^{1,*}

¹Department of Computer Science, University of Pisa

Abstract

Recent advancements in Generative AI have led to the development of systems able to generate text that is often indistinguishable from human-written content. Although this capability has many beneficial applications, it also enables malicious actors to generate synthetic content for deceptive purposes (e.g., proliferation of fake news). As a direct response to all these needs DeSeGma-IT is introduced in the EVALITA 2026 campaign. This challenges the independent and identically distributed assumption (IID), since the generators used to create the training set are different from those used to generate the test set.

DeSegMa is structured into two sub-tasks. Machine Generated Text (MGT) Detection which involves the detection of machine-generated texts at the document level and Human-Machine Text (HMT) Segmentation which involves the segmentation of a document into the human-written and the machine-generated part.

For the first sub-task, DistilBERT is chosen because it has fewer parameters and it is faster than BERT, but it keeps nearly all the language understanding capabilities. For the second sub-task, Light Gradient Boosting Machine (Light GBM) is chosen because it is faster than Gradient Boosting Decision Tree (GBDT) while achieving almost the same accuracy.

The results demonstrated that both models maintain stable performance when moving from validation data to unseen test data.

Keywords

DistilBERT, MGT Detection, LightGBM Regressor, HMT Segmentation.

1. Introduction

Recent advancements in Generative AI and Large Language Models have led to the development of systems able to generate text that is often indistinguishable from human-written content. Although this capability has many beneficial applications, it also enables malicious actors to generate synthetic content for deceptive purposes (e.g., proliferation of fake news).

Beyond public opinion concerns, the European Commission will enforce compliance with the AI Act. In particular, article 50 requires that text generated by General Purpose AI have to be marked and automatically recognizable. Additionally, while state-of-the-art MGT detectors have reported high accuracy, such results often stem from unrealistic experimental settings.

As a direct response to all these needs, DeSeGma-IT - Detection and Segmentation of Machine Generated Text in Italian - [1] is introduced in the EVALITA 2026 campaign [2]. In fact, it challenges the independent and identically distributed assumption (IID), since the generators used to create the training set are different from those used to generate the test set.

DeSegMa is structured into two sub-tasks. SubTask A - Machine Generated Text (MGT) Detection in the Wild - involves the detection of machine-generated texts at the document level, it is structured as a binary-classification problem and it is defined as follows:

Given a piece of text, assign it the label 0, if the text is written by a human, and 1 otherwise.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26-27, Bari, IT

*First author.

✉ niclaauletta.na@gmail.com (N. Auletta)

🌐 <https://github.com/NiclaAuletta> (N. Auletta)

in <https://www.linkedin.com/in/https://www.linkedin.com/in/nicla-auletta/> (N. Auletta)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

SubTask B - Human-Machine Text (HMT) Segmentation - involves the segmentation of a document into the human-written and the machine-generated part, it focuses on localization and it is defined as follows:

Given a piece of text, return the index of the first character that is generated by an LLM [1].

2. Dataset

2.1. Data Understanding

For SubTask A the features of the dataset are text - the text to classify - and label - the ground truth, where 0 means that the text is human written and 1 that it is machine generated.

For SubTask B the features of the dataset are human - the first part of the text, written by a human -, llm - the machine-generated continuation of the text - and human_len - the index of the first character outputted by the language model [1].

2.2. Data Preparation

First, duplicate rows are dropped from both the datasets because with them, the model might incorrectly learn that these instances are much more common or significant than they are.

For SubTask A, only technical noise that do not help detect whether a text is a MGT are removed.

In fact, HTML tags and markdown artifacts, like # and * are used to represent formatting, while URLs are references.

Moreover, emojis add context, emotion, and nuance, but they are the very things computers find hard to quantify and here the goal does not concern sentiment.

Finally, duplicate whitespaces should be normalized to a single space to ensure consistent formatting.

Instead, characters that may be decisive for detection are not removed. For instance, stopwords provide crucial grammatical structure, while punctuation marks may or may not offer valuable information based on the context. Stylometric features, including punctuation marks effectively enhance AI-generated text detection [3].

For SubTask B, nothing is removed from the text before executing this code because it is needed to return the index of the first character from which the llm part starts. Any preprocessing would shift offsets, breaking the mapping between the original text and the computed embeddings. So, the only step done is to concatenate the human and the llm texts.

3. Description of the system

3.1. Subtask A

3.1.1. Model Selection and Architecture

For the classification task, DistilBERT [4] is chosen. It is a compact transformer-based architecture derived from BERT via knowledge distillation. This model is selected for its favourable trade-off between computational efficiency and performance, as it keeps 97% of the language understanding capabilities, while being 60% faster and 40% smaller in parameter count [4]. Specifically, the uncased version is chosen to ensure robustness across varying capitalization styles¹.

This model is primarily aimed at being fine-tuned on tasks that require the whole textual content, as is the case for DeSegMa subtask A. Given its ability to effectively capture the subtle semantic and syntactic

¹<https://huggingface.co/distilbert/distilbert-base-uncased>

patterns critical for differentiating between machine-generated text (MGT) from human-written text (HWT) [5], DistilBERT serves as an ideal back bone for this sequence-level classification task.

3.1.2. Data Processing and Experimental Setup

Textual inputs were tokenized and truncated to the model's maximum sequence length. Dynamic padding is implemented during batch collation, ensuring that sequences were padded only to the length of the longest element within each batch. In fact, this is more efficient with respect to padding the whole dataset to the maximum length [6].

For model validation and hyperparameter tuning, the original training corpus was partitioned to reserve 10% of the data as an independent validation set. This split ensures a reliable assessment of generalization and mitigates the risk of overfitting during the fine-tuning process.

3.1.3. Hyperparameter tuning

To improve the accuracy and the generalization of the model hyperparameter search is run through Bayesian optimization, due to its balance between exploitation (choosing values that performed well previously) and exploration (trying new values to discover better configurations) and its greater efficiency with respect to grid and random search.

As the dataset size grows and models get more complex, this process becomes more expensive since the evaluation of a single set of hyperparameters often requires retraining a new model. Moreover, although the validation error decreases as the training dataset size increases, the ranking of hyperparameter configurations is still relatively consistent across dataset subsets. So, to make this process faster, the training and the validation set previously obtained are subsampled by taking from them respectively 200 and 100 samples. This search is done for 20 trials, based on accuracy, the score used for the evaluation [1], on the following hyperparameters:

- **Learning rate** is a measure of how fast the machine learning model performs data training. With a high learning rate, then your system will rapidly adapt to new data, but it will also tend to quickly forget the old data. With a low learning rate, the system will learn more slowly, but it will also be less sensitive to noise in the new data.
- **Number of train epochs** is the number of training iterations over the dataset [7]. With few epochs, the system can save computation, but it also means we have little training information. With many epochs, the networks can converge during evaluation [8].
- **Per device train batch size** refers to the number of training examples used in one iteration on training process [9]. A small batch size ensures that each training iteration is very fast. Although a large batch size will give a more precise estimate of the gradients, in practice this does not matter much since the direction of the true gradients do not point precisely towards the optimum [7].
- **Weight decay** works by adding a penalty for large weights to the loss function, encouraging the model to learn more generalizable features. Setting it too high can make it difficult for the model to capture important patterns in the data. Setting it too low may not provide sufficient regularization. [10].
- **Warmup ratio** is the ratio of total training steps where learning rate will warm up. Learning rate warm-up is a technique used to stabilize the initial training phase. This allows the model to avoid the unstable gradients that can occur at the beginning of training, particularly with larger models [11].

In Table 1, it is possible to see the hyperparameters search space and the best hyperparameters set. Notice that for the number of train epochs are included only the integers in the interval, while for the learning rate and for the last two hyperparameters also the floats are taken in account. Additionally, for the learning rate the value is sampled from the range in a logarithmic scale, while for the other hyperparameters the linear domain is used.

Table 1

SubTask A: Hyperparameters Search Space and Best Hyperparameters Set

Hyperparameters	Search Space	Best
learning rate	[1e-6, 5e-5]	2.336911157693361e-05
number of train epochs	[2,5]	5
per device train batch size	{4, 8}	4
weight decay	[0.0, 0.3]	0.20388579400869386
warmup ratio	[0.0,0.2]	0.18924653336093455
ACCURACY		0.92

3.2. Subtask B

3.2.1. Model Selection and Architecture

For the segmentation task, Light Gradient Boosting Machine (LightGBM) Regressor is chosen². LightGBM [12] is a newer version of Gradient Boosting Decision Tree Algorithm (GBDT). This model is selected because it speeds up the training process of GBDT by up to 20 times while achieving almost the same accuracy [12].

In particular, the implementation of Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) alongside efficient split finding techniques marks LightGBM as a highly efficient and effective algorithm for regression problems [13]. Since Subtask B requires to map to the target numerical values (the index of the character where the machine generated part of text starts), LightGBM Regressor would be a good back bone for this segmentation task.

3.2.2. Data Processing and Experimental Setup

Textual inputs are encoded by using a Sentence-BERT model [14] as a frozen feature extractor. More precisely, all-MiniLM-L6-v2³ is used because it is a more compact version of the BERT model from the MiniLM (Mini Language Model) and it retains much of the semantic and linguistic understanding of larger transformer models. MiniLM, being trained by using the contrastive learning approach, encourages the mapping of semantically similar sentences closer together in vector space [15].

For model validation and hyperparameter tuning, the original training corpus was partitioned to reserve 20% of the data as an independent validation set. This split provides a large chunk of data for the model to learn from while still reserving a reasonable amount for evaluation.

3.2.3. Hyperparameters tuning

As for Subtask A, hyperparameters search is run through Bayesian optimization. To make this process faster, the training set is subsampled selecting a subset with 20% of the samples, that is 16% of the provided training set.

Moreover, k-fold cross-validation with 3 folds is implemented on this subset to evaluate a model's performance and make sure that it is not dependent on a particular training-test split of the data.

Finally, early stopping is activated to stop training if the model does not improve for a certain number of rounds, saving time and avoiding overfitting [16]. For hyperparameter search, the number of stopping rounds is set to 50, while for training it is set to 100. This search is run for 50 trials, based on MAE (Mean Absolute Error), the score used for the evaluation [1], on the following hyperparameters:

²<https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html>

³<https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

- **Number of leaves** specifies the maximum number of leaves in each tree.
The higher the value, the better the model's performance will be, with a trade-off of having a longer computational time. However, there is a threshold where the impact of adding more leaves will not have much additional impact on the model's performance or even have a negative impact due to overfitting [17].
- **Max depth** specifies the maximum depth of each decision tree.
A lower value can help us prevent overfitting. However, a value that's too low can lead to an underfitting problem.
- **Learning rate** used for SubTask A.
- **Number of estimators** controls the number of trees in the ensemble.
Generally, more trees are better. However, a point of diminishing returns is often reached, and overfitting occurs when there are too many trees [18].
- **Min child samples** specifies the minimum number of samples required in the leaf nodes.
A higher value can help us prevent overfitting. However, a value that is too high can lead to an underfitting problem.
- **Subsample** (bagging fraction) controls what fraction of the data is used for each training iteration. **Column sample by tree** (feature fraction) controls the fraction of features (columns) to consider when building each tree.
Lower values of these last two hyperparameters can help us prevent overfitting and lower the computational time. However, a value that is too low can lead to an underfitting problem [17].
- **Regularization alpha** and **regularization lambda** help contain overfitting by penalizing the excessive weights assigned to certain tree splits.
These elements act as internal brakes, ensuring the model does not excessively memorize the training data [19].

In Table 2, it is possible to see the hyperparameters search space and the best hyperparameters set. Notice that for the all the hyperparameters the linear domain is used. Additionally, for the number of leaves, max depth, number of estimators, min child samples are included only the integers in the interval, while for the others also the floats are taken in account.

Table 2
SubTask B: Hyperparameters Search Space and Best Hyperparameters Set

Hyperparameters	Search Space	Best
number of leaves	[20, 150]	104
max depth	[3, 20]	7
learning rate	[0.01, 0.3]	0.025883831670545893
number of estimators	[100, 2000]	662
min child samples	[5, 100]	40
subsample	[0.5, 1.0]	0.8759855365611233
column sample by tree	[0.5, 1.0]	0.73998086648529
regularization alpha	[0, 1]	0.03013121841344469
regularization lambda	[0, 1]	0.6016304218749495
MAE		100.2806

4. Results

Finally, once the models are trained on their respective training sets, using the validation set for evaluation, they are run on the corresponding test sets and the resulting predictions are submitted.

For SubTask A, the model achieved an accuracy of 0.99, indicating that the model learned the training distribution extremely well. On the unseen test set, the model obtained an accuracy equal to 0.92. This

drop is expected and indicates that the model encounters data that differs slightly from the training distribution. This score therefore represents the true generalization performance of the system.

The high-test accuracy indicates that the model successfully captures linguistic differences between human and machine generated text within the dataset [20]. This system gained the 4th position in the leaderboard shown in Table 3 [1].

Table 3
SubTask A: Leaderboard

Position	Team Name	Accuracy
1	Gradient Descenders	0.945755
2	Kenji Endo	0.942649
3	UniTor	0.928774
4	Nicla	0.924348
5	Stochastic Gradient Descenders	0.921638

For SubTask B, the model achieved a MAE of 98.15 reflecting the model’s ability to learn the structural and linguistic cues present in the training corpus. On the unseen test set, the model obtained a MAE of 102.04. This drop is expected and suggests that the unseen test data introduces patterns or variability not fully captured during training. This score suggests that the model generalizes reasonably well beyond the training distribution.

The results demonstrated that both models maintain stable performance when moving from validation data to unseen test data. This system gained the 5th position in the leaderboard shown in Table 4 [1].

Table 4
SubTask B: Leaderboard

Position	Team Name	Accuracy
1	Stochastic Gradient Descenders	52.54
2	MINDS	56.53
3	Gradient Descenders	62.66
4	UniTor	81.60
5	Nicla	102.04

5. Discussion

The systems we propose ranked 4th and 5th in Subtask A and Subtask B respectively, showing that they can perform the tasks of detecting and segmenting MGT but they can be improved further.

For SubTask A, accuracy alone does not fully characterize performance. Additional metrics would provide deeper insight into class specific behaviour. It is important to note that MGT Detection is sensitive to dataset composition. Detectors struggle in it if the texts come from a domain, a generator, or a language that the model has not seen during training [21].

Therefore, although the model performs well on the provided test set, further evaluation on out of distribution data is necessary to assess robustness. As large language models evolve rapidly, maintaining high detection performance in real world scenarios remains a challenging task.

For SubTask B, the results indicate that the model is able to approximate the transition point between human and machine generated text with reasonable precision. The task itself is challenging, as the boundary between human and machine text does not always correspond to clear lexical or syntactic shifts, making precise prediction difficult [22].

The model's performance may also be influenced by the composition of the training data. If the dataset contains limited stylistic diversity, the model may learn patterns that do not generalize perfectly to new domains or generators [23]. Future work could focus on expanding the dataset, incorporating more diverse writing styles, or exploring architectures specifically designed for boundary detection.

Declaration on Generative AI

During the preparation of this work, the author used:

- **Reverso** to Text Translation
- **Copilot** to Drafting content, Content enhancement, Paraphrase and reword, Citation management.
- **Chat GPT** to Drafting content, Content enhancement.

After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] S. Fariello, G. Fenza, F. Forte, M. Gallo, M. Marotta, Distinguishing Human from Machine: A Review of Advances and Challenges in AI-Generated Text Detection, International Journal of Interactive Multimedia and Artificial Intelligence (2025). URL: https://www.researchgate.net/publication/387246195_Distinguishing_Human_From_Machine_A_Review_of_Advances_and_Challenges_in_AI-Generated_Text_Detection.
- [4] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT: A Distilled Version of BERT: smaller, faster, cheaper and lighter, in: EMC²: 5th Edition Co-located with NeurIPS 2019, 2020. URL: <https://www.emc2-ai.org/assets/docs/neurips-19/emc2-neurips19-paper-33.pdf>.
- [5] T. O. Abiola, T. A. Bizuneh, O. J. Abiola, T. O. Oladepo, O. E. Ojo, G. Sidorov, O. Kolesnikova, CIC-NLP at GenAI detection task 1: Leveraging DistilBERT for detecting machine-generated text in English, in: Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect), 2025. URL: <https://aclanthology.org/2025.genaidetect-1.29.pdf>.
- [6] HuggingFace, Text classification, 2022. URL: https://huggingface.co/docs/transformers/tasks/sequence_classification.
- [7] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly Media, 2022. URL: <https://0-lucas.github.io/digital-garden/99.-Books/Hands-on-Machine-Learning.pdf>.
- [8] R. Egele, F. Mohr, T. VIERING, P. Balaprakash, The unreasonable effectiveness of early discarding after one epoch in neural network hyperparameter optimization, Neurocomputing (2024). URL: <https://www.sciencedirect.com/science/article/pii/S0925231224007355>.
- [9] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016. URL: <http://www.deeplearningbook.org>.
- [10] P. Iusztin, N. Sen, J. Chaumond, H. Tahir, A. Gulli, J. Chaumond, H. Tahir, A. G. Gulli, LLM Engineer's Handbook, Sciendo, 2024. URL: <https://doi.org/10.0000/9781836200062>.
- [11] A. Sarkar, Deep Learning Dynamics: The Science Behind AI Training: Exploring the Strategies, Challenges, and Innovations Shaping Modern AI Development, Sarkar, A., 2025.

- [12] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: *Advances in Neural Information Processing Systems*, 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.
- [13] C. Kahraman, S. C. Onar, S. Cebi, B. Oztaysi, A. C. Tolga, I. U. Sari, *Intelligent and Fuzzy Systems: Intelligent Industrial Informatics and Efficient Networks Proceedings of the INFUS 2024 Conference, Volume 2*, Springer Nature Switzerland, 2024.
- [14] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL: <https://aclanthology.org/D19-1410/>.
- [15] H. Arabnia, L. Deligiannidis, F. Shenavarmasouleh, S. Amirian, F. Mohammadi, *Computational Science and Computational Intelligence: 11th International Conference, CSCI 2024, Las Vegas, NV, USA, December 11–13, 2024, Proceedings, Part V*, Springer Cham, 2025.
- [16] G. Kunapuli, *Ensemble Methods for Machine Learning*, Manning, 2023.
- [17] L. Owen, *Hyperparameter Tuning With Python: Boost your machine learning model’s performance via hyperparameter tuning*, Packt Publishing, 2022.
- [18] A. van Wyk, *Machine Learning with LightGBM and Python A Practitioner’s Guide to Developing Production-ready Machine Learning Systems*, Packt Publishing, 2022.
- [19] D. Rodrigues, *LEARN LIGHTGBM: Build Accurate Models with Scalable Machine Learning*, StudioD21, 2025.
- [20] A. Masih, B. Afzal, S. Firdoos, J. Mahmood, A. Ali, M. S. Abdulnabi, D. M. Balungu, Classifying human vs. AI text with machine learning and explainable transformer models, *Scientific Reports* (2025). URL: <https://www.nature.com/articles/s41598-025-27377-z>.
- [21] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. Mohammed Afzal, T. Mahmoud, T. Sasaki, T. Arnold, A. F. Aji, N. Habash, I. Gurevych, P. Nakov, M4: Multi-generator, Multi-domain, and Multi-lingual Black-Box Machine-generated Text Detection, in: *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL: <https://aclanthology.org/2024.eacl-long.83/>.
- [22] L. Kushnareva, T. Gaintseva, G. Magai, S. Barannikov, D. Abulkhanov, K. Kuznetsov, E. Tulchinskii, I. Piontkovskaya, S. Nikolenko, AI-generated Text Boundary Detection with Roft, in: *1st Conference on Language Modeling (COLM) 2024*, 2024. URL: <https://arxiv.org/pdf/2311.08349>.
- [23] K. Jiao, Q. Wang, L. Zhang, Z. Guo, Z. Mao, M-Rangedetector: Enhancing Generalization in Machine-Generated Text Detection through Multi-Range Attention Masks, in: *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. URL: <https://aclanthology.org/2025.findings-acl.469.pdf>.

Stochastic Gradient Descenders at DeSegMa-IT: Instruction-Tuned LLM and Token Classification for MGT Detection and Boundary Localization

Huynh Nguyen Phu^{1,2}, Bui Hong Son^{1,2} and Dang Van Thin^{1,2}

¹University of Information Technology, Ho Chi Minh City, Vietnam

²Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

We describe our top-ranked system for the DeSegMa-IT shared task (EVALITA 2026), which achieved **1st place** in Subtask B (boundary localization) with an MAE of 52.54. Our key insight is to reframe boundary detection as a *token classification* problem rather than direct character-offset regression: by labeling each token as human- or machine-written, we enable a monolingual Italian encoder (UmBERTo) to capture fine-grained stylistic shifts and pinpoint the transition point. Unlike sequence-level classifiers or generative approaches that predict offsets end-to-end, our token-level formulation provides richer supervision and naturally handles variable-length documents. For Subtask A (binary classification), we fine-tuned Qwen2.5-0.5B-Instruct with LoRA, achieving Rank 5 (Accuracy 0.92).

1. Introduction

The proliferation of Large Language Models (LLMs) capable of generating high-quality text has created an urgent need for robust detection systems. This is particularly critical in scenarios involving distribution shifts, where detectors trained on one domain must generalize to unseen topics or prompting strategies. DeSegMa-IT [1], a shared task at EVALITA 2026 [2], addresses these challenges by benchmarking systems on two complementary problems: document-level detection of machine-generated text (MGT) and fine-grained segmentation of human-machine mixed content.

The shared task is divided into two subtasks:

- **Subtask A (Binary Classification):** Determining whether a full document was written by a human or generated by a machine.
- **Subtask B (Boundary Localization):** Identifying the exact character index where a human-written prefix transitions into a machine-generated continuation.

Our work investigates two key research questions guiding our dual-system design: (1) Can reformulating boundary localization as token classification—rather than direct offset regression—yield more accurate transition-point detection by leveraging fine-grained stylistic signals? (2) How effectively can parameter-efficient fine-tuning of a lightweight instruction-tuned LLM generalize to unseen MGT scenarios with limited computational resources? To address these, we make three main contributions:

- **Reformulate** the boundary localization task as dense token classification, enabling token-level supervision that captures local stylistic shifts between human and machine text.
- **Achieve state-of-the-art** performance on Subtask B (**Rank 1**, MAE 52.54), demonstrating that token-level formulation outperforms regression-based and sequence-level baselines.
- **Demonstrate** competitive binary classification (Rank 5, Accuracy 0.92) using LoRA-tuned Qwen2.5-0.5B-Instruct with only 0.5B parameters—an order of magnitude smaller than typical LLM-based detectors.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26–27, Bari, IT

✉ 24521352@gm.uit.edu.vn (H. N. Phu); 22521246@gm.uit.edu.vn (B. H. Son); thindv@uit.edu.vn (D. V. Thin)

🆔 0009-0004-6867-5853 (H. N. Phu); 0009-0006-7420-9212 (B. H. Son); 0000-0001-8340-1405 (D. V. Thin)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Table 1

Dataset statistics for both subtasks.

Subtask	Metric	Train	Dev	Test
A	Total samples	33,138	5,059	15,135
	Human (label=0)	16,569	—	—
	Machine (label=1)	16,569	—	—
B	Total samples	19,945	4,579	26,015

- **Provide** empirical analysis of error patterns and boundary-offset distributions, offering insights for future MGT detection research.

In this paper, we describe the system submitted by the *Stochastic Gradient Descenders* team. Our approach prioritizes specialized architectural choices for each subtask rather than a monolithic solution. For Subtask A, we employ a prompt-based learning strategy, fine-tuning a lightweight instruction-following model (Qwen2.5-0.5B-Instruct [3]) via Low-Rank Adaptation (LoRA) [4]. This allows us to leverage the reasoning capabilities of LLMs while maintaining high computational efficiency. For Subtask B, we treat segmentation as a dense token-labeling problem, utilizing a fully fine-tuned Italian encoder (UmBERTo [5]) to capture local stylistic shifts.

The remainder of this paper details our methodology (Section 3), experimental setup (Section 4), and a discussion of the results and error analysis (Section 6).

2. Task and Dataset Description

DeSegMa-IT [1] is a shared task at EVALITA 2026 that benchmarks machine-generated text (MGT) detection systems on Italian text.

2.1. Task Definitions

The shared task comprises two subtasks:

- **Subtask A (Binary Classification):** Given a document, predict whether it is entirely human-written (label 0) or machine-generated (label 1). Evaluation metric: Accuracy.
- **Subtask B (Boundary Localization):** Given a mixed document (human prefix + LLM continuation), predict the 0-based character index where the transition occurs. Evaluation metric: Mean Absolute Error (MAE).

2.2. Dataset Statistics

Table 1 summarizes the dataset splits for both subtasks. Subtask A contains 33,138 training samples with a perfectly balanced label distribution (50% human, 50% machine). Subtask B provides 19,945 training samples of human-machine hybrid documents.

2.3. Text Length Analysis

Subtask A. Documents exhibit high variance in length: mean 2,423 characters (median 1,697, 95th percentile 5,525). This motivates our use of truncation-aware prompting for the LLM-based classifier.

Subtask B. Mixed documents are relatively uniform in length (mean 1,416, std 302 characters). The boundary position (human_len) averages 265 characters with a mean ratio of 0.20 relative to document length, indicating that most transitions occur in the first quarter of the text. This skewed distribution informed our token-level labeling strategy, which can capture fine-grained transitions regardless of absolute position.

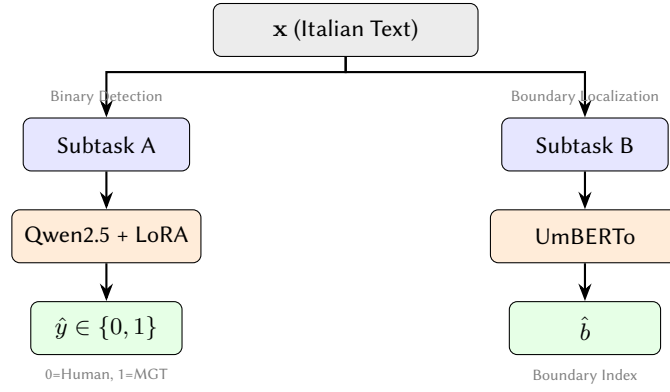


Figure 1: System overview. Our approach employs two independent pipelines: an instruction-tuned LLM for binary detection (Subtask A) and a token classifier for boundary localization (Subtask B).

3. Methodology

This section details the system architecture and training pipelines employed for the DeSegMa-IT shared task. We present our approaches for Subtask A and Subtask B sequentially, as illustrated in Figure 1.

3.1. Subtask A: Binary Detection via Instruction Tuning

Problem Formulation

We approach Subtask A as a conditional generation problem using a decoder-only Large Language Model (LLM). Rather than initializing a randomly weighted classification head on top of the backbone, we leverage the model’s pre-existing instruction-following capabilities. The task is framed as a single-token generation constraint: the model is prompted to output a specific digit representing the class label (see Figure 2).

Data Formatting and Supervision

For each input document, we construct a conversation-style prompt adhering to the *ChatML* format. The system instruction is consistent across all instances:

“You are a binary MGT detector. Return exactly one digit: 0 for human-written, 1 for machine-generated.”

The user message contains the raw document text. During training, the assistant’s response is the ground-truth label (“0” or “1”). We employ standard supervised fine-tuning (SFT) with separate masking: the cross-entropy loss is computed exclusively on the assistant’s response tokens, while the system and user instructions are masked out.

Backbone and Parameter-Efficient Adaptation

Our training infrastructure for Subtask A ($2 \times$ NVIDIA T4 on Kaggle) constrains the backbone to models under 3B parameters. Within this budget, we experimented with two sizes from the Qwen2.5-Instruct family [3]: the 0.5B and 3B variants. In preliminary trials, the 0.5B model achieved higher validation accuracy than its 3B counterpart, which we attribute to the larger model’s tendency to overfit under LoRA fine-tuning with limited training epochs. We therefore select Qwen2.5-0.5B-Instruct as our backbone, which also benefits from strong multilingual coverage (including Italian) and robust instruction-following abilities at this compact scale. To mitigate catastrophic forgetting and reduce computational overhead, we apply Low-Rank Adaptation (LoRA) [4] using the `unsloth` library for optimized training [6]. Our LoRA configuration targets all linear layers with rank $r = 16$, $\alpha = 32$, and dropout rate 0.05. We train with 16-bit precision (mixed precision) to preserve dynamic range, avoiding 4-bit quantization to maximize performance given the manageable model size.

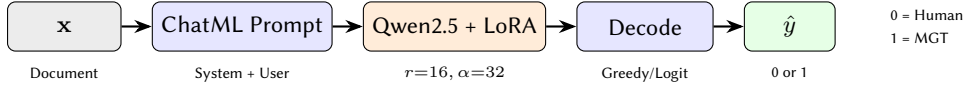


Figure 2: Instruction-tuning pipeline for Subtask A. The document is formatted as a ChatML prompt, processed by Qwen2.5 with LoRA adaptation, and decoded to predict the binary label $\hat{y}$.

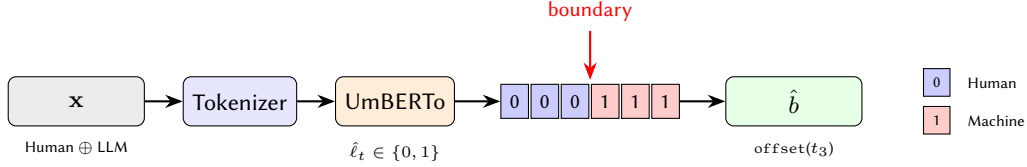


Figure 3: Token classification pipeline for Subtask B. Each token receives a binary label, and the boundary $\hat{b}$ is the offset of the first machine-labeled token.

Training Protocol

We allocate an 80/20 stratified split of the provided training set for internal validation. Training is performed for 2 epochs with a maximum sequence length of 1024 tokens. We use a per-device batch size of 4 with gradient accumulation steps set to 5, resulting in an effective batch size of 20. The optimizer is AdamW [7] (8-bit quantization [8]) with a learning rate of 1×10^{-4} , a warmup ratio of 0.05, and a weight decay of 0.01. We disable the `use_cache` configuration to support gradient checkpointing during training.

Inference Strategy

At inference, we set `max_new_tokens=2` as a safety buffer: the model is expected to produce a single digit followed by the EOS token, but occasionally emits a whitespace or newline character before the digit. Allowing two tokens ensures the digit is captured even when preceded by such spurious characters. The generated string is stripped of whitespace and only its first character is inspected, so the second token slot is never used for classification. To ensure robustness, we implement a two-step fallback mechanism:

1. **Greedy Decoding:** The stripped output is checked; if its first character is “0” or “1”, it is accepted as the prediction.
2. **Logit Fallback:** In rare cases where neither character appears (e.g., the model produces only special tokens), we perform a separate forward pass on the original prompt and inspect the logits at the *last prompt position*—i.e., the distribution over the *first* token the model would generate. The class whose token ID receives the higher logit (“0” vs. “1”) is selected.

3.2. Subtask B: Boundary Localization via Token Classification

Token-Level Formulation

While Subtask B is evaluated as a regression task (predicting a character index), we reformulate it as a dense sequence labeling problem. Given a document x composed of a human-written prefix and a machine-generated continuation, we tokenize the sequence and assign a binary label ℓ_t to each token t :

$$\ell_t = \begin{cases} 0 & \text{if } \text{start_offset}(t) < \text{boundary_index} \\ 1 & \text{otherwise} \end{cases}$$

This simple rule assigns a token to the human class if it *begins* before the boundary, effectively treating tokens that straddle the boundary as human. The formulation provides dense supervision signals compared to a single scalar regression target, allowing the model to learn local textural shifts (see Figure 3).

Backbone Model

We fine-tune `Musixmatch/umberto-commoncrawl-cased-v1` (UmBERTo) [5], a RoBERTa-based [9] model pre-trained on a large Italian corpus. A standard token classification head is initialized on top of the encoder to project hidden states into the binary label space.

Training Protocol

The official training data is split 80/20 for training and validation. We tokenize inputs with a maximum length of 512 tokens using truncation. Crucially, we maintain full offset mappings to allow precise reconstruction of character indices during inference. The model is fully fine-tuned (all parameters) for 10 epochs using mixed precision (FP16). We use a batch size of 8 per device with gradient accumulation 2 (effective batch size 16), a learning rate of 2×10^{-5} , linear warmup of 0.1, and weight decay of 0.01. Evaluation is performed every 250 steps to monitor the Mean Absolute Error (MAE) on the validation set.

Decoding Strategy

Inference involves a two-step process:

1. **Prediction:** The model outputs per-token class probabilities. We take the argmax to obtain a sequence of binary labels.
2. **Boundary Extraction:** We iterate through the predicted labels to find the first token classified as ‘machine’ ($\hat{\ell}_t = 1$). The start character offset of this token is returned as the predicted boundary. If no token is classified as machine, the boundary is set to the document length.

This strategy aligns with the task definition (identifying the *start* of the generated text) while leveraging the contextual power of the bidirectional encoder.

4. Experimental Setup

4.1. Implementation Details

All experiments were implemented using the PyTorch framework [10] and the Hugging Face ecosystem [11]. For Subtask A, we utilized the `unsloth` library [6] to accelerate the LoRA fine-tuning of Qwen2.5, trained on $2 \times$ NVIDIA T4 GPUs provided by Kaggle.¹ For Subtask B, we used the standard Trainer API from the Transformers library, trained mainly on a single NVIDIA H100 (80GB VRAM) provided by FPT AI Factory.² To ensure reproducibility, we fixed the random seed to 42 for all data splits and model initializations.

4.2. Validation Protocols

Subtask A We employed a stratified hold-out validation strategy, reserving 20% of the provided training data. Model checkpoints were evaluated based on classification accuracy and the formatting validity of the generated tokens. Given the stability of the instruction-tuned backbone, we observed rapid convergence; the final submission utilized the checkpoint from the end of the second epoch.

Subtask B For the segmentation task, we used an 80/20 random split of the labeled data. Metrics were computed at the character level (MAE) rather than the token level to align with the official competition metric. During development, we also explored a 5-fold cross-validation scheme to assess the robustness of different encoder backbones before selecting UmBERTo for the final submission.

¹<https://www.kaggle.com>

²<https://ai.fptcloud.com/>

Table 2

Offline validation results for Subtask B. The *Formulation* column refers to the decoding heuristic used.

Backbone	Formulation	MAE
Italian BERT (5-fold CV)	Token Cls + First Offset	55.78
mDeBERTa-v3	Token Cls + First Offset	55.35
UmBERTo	Token Cls + First Offset	53.30
UmBERTo	Token Cls + Global+Biased	52.06

Table 3

Results of progressive method improvements on the development set for Subtask B. Each row adds one component to the previous configuration. Δ indicates the improvement over the previous row.

#	Configuration	MAE	Δ
1	UmBERTo baseline (3 epochs, simple decode)	55.12	—
2	+ Increase to 5 epochs	53.30	−1.82
3	+ Global score decoding	52.89	−0.41
4	+ Length bias tuning ($\beta = -0.15$)	52.06	−0.83
Final system (row 4)		52.06	—

4.3. Baseline Comparisons for Subtask B

In the preliminary phase, we benchmarked several architectures for the boundary localization task:

- **mDeBERTa-v3-base** [12]: A multilingual transformer with 86M backbone parameters (276M total with embeddings), featuring disentangled attention that separately encodes content and position. The model was pre-trained on the CC100 multilingual corpus with a 250K-token vocabulary.
- **Italian BERT** (bert-base-italian-xxl-cased) [13]: A monolingual BERT variant with approximately 110M parameters, pre-trained on OPUS corpora extended with the Italian portion of OSCAR (81GB total). The cased tokenization preserves orthographic cues relevant for stylistic analysis.
- **UmBERTo** (umberto-commoncrawl-cased-v1) [5]: A RoBERTa-based Italian model employing SentencePiece tokenization with a 32K vocabulary and whole-word masking. Pre-trained on the Italian subcorpus of OSCAR, it captures language-specific morphological and syntactic patterns.

All baselines were trained with the same token-classification objective described in Section 3.

4.4. Offline Validation Results

Table 2 presents the offline performance of our Subtask B models on the development splits. UmBERTo consistently outperformed other backbones, likely due to its superior handling of Italian morphology and syntax compared to the multilingual or standard BERT variants. We also experimented with a “global + biased” decoding strategy (tuning a bias term to penalize/reward the positive class), which yielded further improvements in offline settings. However, for the final submission, we opted for the simpler standard decoding to avoid overfitting to the development set distribution. To understand the contribution of each component, we also conducted ablation experiments starting from a vanilla UmBERTo baseline and progressively adding components (Table 3).

We also experimented with several training-time tricks that ultimately did *not* improve the final performance:

Key Findings:

Table 4

Negative results: training tricks that degraded or did not improve performance on the development set.

Configuration	MAE	vs. Baseline
Weighted sampling (extreme cases $\times 3.0$)	56.87	+1.75
Boundary token re-weighting ($k=5$)	54.21	−0.91
Custom regression head ($\lambda=0.1$)	55.43	+0.31
Custom regression head ($\lambda=0.01$)	53.67	−1.45
Boundary weighting + regression ($k=3$, $\lambda=0.01$)	53.95	−1.17

Table 5

Official Subtask A leaderboard [14].

Pos.	Team	Accuracy
1	Gradient Descenders	0.945 755
2	Kenji Endo	0.942 649
3	UniTor	0.928 774
4	Nicla	0.924 348
5	Stochastic Gradient Descenders (Ours)	0.921 638

- **Training epochs:** Extending training from 3 to 5 epochs provided consistent gains, though validation loss plateaued after epoch 5–6.
- **Decoding strategy:** The global score decoding, which considers the joint probability of all tokens rather than the first positive prediction, improved boundary precision.
- **Length bias:** Tuning a small bias term ($\beta = -0.15$) to slightly penalize the machine-generated class probability helped correct for systematic over-prediction of short boundaries.
- **Weighted sampling:** Oversampling extreme boundary lengths (very short or very long human segments) caused over-prediction bias, likely because the model learned to expect extreme cases more frequently than in the test distribution.
- **Auxiliary regression head:** Adding a regression head to directly predict the boundary character index did not consistently help, suggesting that the token classification objective already captures sufficient information.

5. Official Results

We report the official leaderboard results as published by the organizers [14]. Following the paper structure, Subtask A results are presented before Subtask B.

5.1. Subtask A: Binary Classification

Table 5 presents the official Subtask A leaderboard, where our system achieves 5th place with an accuracy of 0.9216, trailing the top-performing team (Gradient Descenders) by 2.4 percentage points. The result demonstrates that instruction-tuned LLMs can achieve competitive classification performance with minimal task-specific adaptation. By applying LoRA to Qwen2.5-0.5B-Instruct, we update only a small fraction of the model’s parameters while preserving its pre-trained reasoning capabilities, yet reach accuracy within a reasonable margin of the top-ranked systems. Notably, the top-4 systems cluster within a narrow 2.1-point accuracy band (0.9246–0.9458), which may suggest that the remaining performance gap is increasingly driven by stylistic ambiguity in edge cases rather than by differences in model capacity alone.

Table 6

Official Subtask B leaderboard [14].

Pos.	Team	MAE
2	MINDS	56.53
3	Gradient Descenders	62.66
4	UniTor	81.60
5	Nicla	102.04
1	Stochastic Gradient Descenders (Ours)	52.54

5.2. Subtask B: Boundary Localization

Table 6 presents the official Subtask B leaderboard, where our system achieves **1st place** with an MAE of 52.54 characters—outperforming the runner-up (MINDS) by 3.99 points and the third-place team (Gradient Descenders) by a substantial 10.12 points (16.2% relative improvement). We attribute our strong performance primarily to two design decisions validated in our offline ablations (Section 4): (1) the choice of UmBERTo, a monolingual Italian encoder whose language-specific pre-training yields more reliable token-level predictions than multilingual alternatives, and (2) the simple first-positive decoding heuristic, which avoids introducing additional error through complex post-processing. Furthermore, the steep performance drop-off beyond the top-2 teams (UniTor at 81.60 MAE, Nicla at 102.04 MAE) indicates that boundary localization is a substantially harder task than binary classification, rewarding architectures specifically optimized for fine-grained sequence labeling over general-purpose approaches.

5.3. Summary

The proposed dual-strategy approach achieves competitive performance on Subtask A (5th place, 0.9216 accuracy) and state-of-the-art performance on Subtask B (1st place, 52.54 MAE). These results validate two complementary insights: (1) parameter-efficient LLM adaptation provides a strong baseline for binary detection without extensive fine-tuning, and (2) monolingual encoders with dense token-level supervision excel at fine-grained boundary localization. The contrasting outcomes across subtasks—mid-tier classification but top-tier segmentation—highlight that task-specific architectural choices matter more than raw model scale, particularly for structured prediction problems where language-specific morphological cues provide discriminative signal.

6. Discussion

This section presents a structured error analysis of our systems, identifying failure patterns and linking them to design choices. We support our analysis with confusion matrices and representative error cases.

6.1. Subtask A

Table 7 presents the confusion matrix on our 20% held-out validation set (6,628 samples). The system achieves high overall accuracy (92.2%), but the error distribution reveals asymmetric failure modes. From manual inspection of 100 randomly sampled errors, we identify three dominant patterns: false positives (8.2%) where human-written texts are misclassified as machine-generated, primarily when the writing is highly formal or follows templated structures such as press releases and legal notices; false negatives (7.4%) where machine-generated texts exhibiting high fluency and contextual coherence—particularly advanced LLM outputs mimicking journalistic style—are misclassified as human; and length-induced truncation errors, where the model only observes the initial 1,024 tokens and misses telltale machine artifacts in the continuation.

Two representative cases illustrate these failure modes. The first is a human-written corporate announcement—*“Il Consiglio di Amministrazione, riunitosi in data odierna, ha deliberato l’approvazione*

Table 7

Confusion matrix for Subtask A. Rows = ground truth, columns = predictions.

	Pred: Human	Pred: Machine	Total
Actual: Human	3,042	272	3,314
Actual: Machine	245	3,069	3,314
Total	3,287	3,341	6,628

Table 8

Token-level confusion matrix for Subtask B. Special tokens excluded.

	Pred: Human	Pred: Machine
Actual: Human	876,421	47,892
Actual: Machine	89,341	1,289,102

Table 9

Error analysis by human-prefix length.

Human Prefix	Count	MAE	Bias	Error %
0–100 chars	2,404	74.02	+73.17	9.3%
100–200 chars	6,345	57.13	+44.32	22.1%
200–300 chars	6,394	51.96	+18.14	14.6%
300–400 chars	6,369	47.70	−10.71	12.3%
400–512 chars	4,487	61.92	−52.97	17.8%

del bilancio consolidato per l’esercizio 2025...—misclassified as machine-generated due to its bureaucratic register, passive constructions, and absence of subjective markers. The second is an LLM-generated journalistic text (380 characters)—*“Roma si è svegliata sotto un cielo grigio, ma l’entusiasmo dei tifosi non si è spento. ‘Oggi vinciamo noi’, mi ha detto Marco...*”—misclassified as human-written because it successfully mimics discourse markers such as direct quotations and emotional language.

These failures reveal that the model conflates *stylistic register* with authorship, failing when human writing is highly formal or when LLMs produce contextually rich, conversational text. The instruction-tuning formulation helped by preserving the LLM’s pre-trained language understanding, enabling detection of subtle coherence issues without task-specific feature engineering. However, the 1,024-token truncation and single-pass inference prevent the model from leveraging document-level patterns, and the balanced training made the model overconfident on edge cases.

6.2. Subtask B

Table 8 presents the aggregated token-level confusion matrix across the validation set. With 94.1% token-level accuracy, the model reliably distinguishes human from machine tokens, but errors concentrate near the boundary. Stratifying character-level MAE by human-prefix length (Table 9) reveals a clear centering bias: the model over-predicts boundaries for short prefixes (positive bias up to +73 characters for 0–100 char prefixes) and under-predicts for long ones (negative bias of −53 characters for 400–512 char prefixes).

Analysis of high-error cases identifies three dominant patterns: smooth stylistic transitions (38%) where the LLM seamlessly continues the topic with minimal discontinuity, causing delayed detection; formal human writing with LLM-like features (27%) that triggers premature false positives; and noisy oscillating predictions near the boundary (21%) where the first-positive heuristic selects a premature transition. Two examples demonstrate these patterns. In an over-prediction case (+47 chars), the text *“...il governo ha annunciato nuove misure. [TRUE] Le riforme prevedono un incremento... [PRED]”* shows the LLM maintaining the same formal register, causing the model to delay detection until encountering

more stereotypical machine markers. In an under-prediction case (−32 chars), the text “*Il rapporto dell’ISTAT evidenzia che [PRED] i tassi di disoccupazione [TRUE] hanno raggiunto...*” shows human-authored statistical reporting triggering premature detection due to domain-specific terminology the model associates with LLM outputs.

Both errors stem from the binary token-labeling scheme’s assumption of sharp stylistic discontinuity—an assumption violated when human and machine text share similar registers. Token-level supervision provided dense feedback that enabled fine-grained stylistic pattern learning, and UmBERTo’s monolingual pre-training captured Italian-specific morphological cues (verb conjugations, article agreement) that signal authorship changes. However, the binary labeling assumes a sharp boundary that may not exist in practice, and the first-positive decoding commits to early predictions even under model uncertainty.

6.3. Cross-Task Insights and Takeaways

Comparing errors across both subtasks reveals a shared vulnerability: formal, templated human writing is systematically confused with machine-generated text. This suggests that current detection methods—including ours—rely heavily on stylistic fluency as a proxy for machine authorship, a heuristic that degrades as LLMs improve and as human writing becomes more formulaic.

Our analysis yields three main findings: (1) monolingual encoders outperform multilingual alternatives for Italian MGT detection, likely due to better morphological modeling; (2) token-level supervision is effective, but centering bias and smooth transitions remain challenging; and (3) truncation-based errors are non-negligible, as the first 512–1,024 tokens may not always contain decisive signals. Potential mitigations include length-aware calibration to reduce centering errors, multi-span inference for long documents, and soft boundary modeling to better reflect gradual human-machine transitions. These directions remain future work; our current system prioritizes simplicity and strong baseline performance.

7. Conclusion

This paper presented the *Stochastic Gradient Descenders* submission to the DeSegMa-IT shared task. We proposed a dual-strategy system: a parameter-efficient, prompt-based LLM (Qwen2.5) for binary detection and a dense token-classification approach using a monolingual Italian encoder (UmBERTo) for boundary localization. Our results—ranking **1st** in Subtask B and achieving competitive accuracy in Subtask A—demonstrate the effectiveness of task-specific architectural choices. Specifically, we showed that monolingual models remain a powerful baseline for fine-grained analysis of specific languages, outperforming larger multilingual counterparts in segmentation tasks. Future work will investigate multi-task learning frameworks to jointly address detection and localization, as well as calibration techniques to mitigate length-related biases in segmentation.

Acknowledgments

We thank the DeSegMa-IT organizers for releasing the dataset and maintaining the evaluation platform.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: assist with grammar and spelling checks, paraphrase and reword sentences for clarity, and support code debugging during experiment implementation. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] G. Puccetti, A. Pedrotti, A. Esuli, DeSegMa-IT at EVALITA 2026: Overview of the Detection and Segmentation of Machine Generated Text in Italian Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026, p. to appear.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, EVALITA 2026: Overview of the 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026, p. to appear.
- [3] Qwen Team, Qwen2.5 Technical Report, arXiv preprint arXiv:2412.15115 (2024).
- [4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, arXiv preprint arXiv:2106.09685 (2021).
- [5] Musixmatch Research, UmBERTo: an Italian Language Model trained with Whole Word Masking, <https://huggingface.co/Musixmatch/umberto-commoncrawl-cased-v1>, 2020. Accessed: 2025-12-31.
- [6] UnslothAI, Unsloth: Efficient Fine-tuning and Reinforcement Learning for LLMs, <https://github.com/unslothai/unsloth>, 2024. Accessed: 2025-12-31.
- [7] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: International Conference on Learning Representations (ICLR), 2019, pp. 1–18.
- [8] T. Dettmers, M. Lewis, S. Shleifer, L. Zettlemoyer, 8-bit Optimizers via Block-wise Quantization, arXiv preprint arXiv:2110.02861 (2021).
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, arXiv preprint arXiv:1907.11692 (2019).
- [10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.
- [11] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Transformers: State-of-the-Art Natural Language Processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.
- [12] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with Disentangled Attention, arXiv preprint arXiv:2006.03654 (2020).
- [13] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.
- [14] DeSegMa-IT Organizers, DeSegMa-IT Leaderboard (Results), <https://desegma.github.io/results/>, 2025. Accessed: 2025-12-31.

UniTor at DeSegMa-It: Analyzing Supervision and Encoder Representations for Italian Machine-Generated Text Detection

Federico Borazio^{*1}, Giacomo De Luca^{*1}, Daniele Pasquini^{*1}, Danilo Croce¹ and Roberto Basili¹

¹Department of Enterprise Engineering, University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy

^{*}Equal contribution.

Abstract

This paper describes the UniTor systems submitted to the DeSegMa-IT shared task at EVALITA 2026. Our study investigates the impact of unsupervised versus supervised approaches for machine-generated text (MGT) detection and segmentation in Italian, and analyzes which components of encoder-only architectures are most relevant for these tasks. For Subtask A, we compare a data-agnostic, zero-shot detector based on Fast-DetectGPT with supervised ModernBERT-based classifiers fine-tuned on Italian data. For Subtask B, we address boundary localization via token-level classification followed by change point detection, evaluating both supervised transformer-based models and an unsupervised token-level adaptation of Fast-DetectGPT. While our submitted systems achieve mid-range leaderboard performance, post-submission experiments reveal a key divergence: for document-level detection, model size proves decisive, with the English-centric ModernBERT-large outperforming the MultilingualModernBERT-base alternative; for boundary localization, multilingual pretraining yields substantial gains, reducing error by over 25%. These results explore the tradeoffs between robustness and task-specific supervision, and show that document-level detection relies on structural features that generalize across languages, whereas token-level segmentation requires language-specific models.

Keywords

Machine Text Detection, Encoder Models, Layer Probing, Fine-Tuning, Zero-Shot Detection

1. Introduction

The rapid evolution of Large Language Models (LLMs) has democratized the ability to generate text that is increasingly difficult to distinguish from human writing. While this technological progress enables a wide range of beneficial applications, it has also substantially lowered the cost and expertise required to produce large volumes of synthetic content, facilitating its use in deceptive, manipulative, or low-quality information campaigns. Recent state-of-the-art open-weight models, such as DeepSeek [1] and Kimi [2], provide high-quality text generation at scale, enabling the automatic production of fluent and coherent narratives across a wide variety of domains. As a consequence, machine-generated text (MGT) is no longer confined to clearly artificial or low-effort outputs, but is often embedded seamlessly within otherwise legitimate content streams. This phenomenon has attracted increasing public attention, as reflected by the designation of “slop” as the Word of the Year for 2025 by the Merriam-Webster dictionary¹.

At the same time, recent advances in model alignment manipulation have further complicated the detection landscape. Uncensored or weakly aligned variants of open-source models are increasingly accessible, either through architectural interventions (e.g., ablation of refusal mechanisms) or post-training fine-tuning procedures. Prior work has shown that refusal behavior in aligned LLMs can

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ borazio@ing.uniroma2.it (F. Borazio^{*}); deluca@ing.uniroma2.it (G. De Luca^{*}); daniele.pasquini@uniroma2.eu (D. Pasquini^{*}); croce@info.uniroma2.it (D. Croce); basili@info.uniroma2.it (R. Basili)

🆔 0009-0000-0193-2131 (F. Borazio^{*}); 0009-0004-5469-3364 (G. De Luca^{*}); 0009-0006-8172-3228 (D. Pasquini^{*}); 0000-0001-9111-1950 (D. Croce); 0000-0001-5140-0694 (R. Basili)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://www.merriam-webster.com/wordplay/word-of-the-year>

be mediated by low-dimensional subspaces in the residual stream and can therefore be substantially altered without significantly degrading generation quality [3]. As a result, machine-generated texts can easily include stylistic features, such as profanity, idiosyncratic phrasing, or unconventional discourse patterns, that were previously strong cues of human authorship.

These developments pose a critical challenge for existing detection systems. Detectors trained primarily on RLHF-aligned models and conventional prompting strategies often exhibit limited robustness to paraphrasing [4], adversarial rewriting [5], or stylistic deviations introduced by weakly aligned generators. Consequently, reliable MGT detection must operate under realistic, non-I.I.D. conditions in which both the generating models and their alignment properties are unknown.

While the proliferation of synthetic content is a global phenomenon, defensive mechanisms are hindered by a severe asymmetry in available linguistic resources. A central bottleneck lies in the overwhelming predominance of English in both detection benchmarks and training corpora. Foundational datasets for MGT detection, such as TuringBench [6] and the HC3 corpus [7], are almost exclusively English-centric, reflecting a broader trend in large-scale language modeling.

Although recent evaluation campaigns have introduced multilingual components [8, 9], languages other than English remain substantially underrepresented. For Italian in particular, detection strategies often rely on translated data or small ad-hoc datasets, which fail to capture language-specific syntactic, morphological, and discourse-level artifacts [10]. This resource gap creates a critical blind spot: without native benchmarks that simulate adversarial “in the wild” conditions, the robustness and reliability of state-of-the-art MGT detectors for Italian remain largely unverified.

To address these challenges within the Italian landscape, this study targets the **DeSegMa-IT** [11] shared task in Evalita 2026 [12]. DeSegMa-IT is designed to benchmark the robustness of systems against realistic, non-I.I.D. scenarios and is structured into two main sub-tasks:

- **Subtask A: MGT Detection in the Wild.** This subtask is formulated as a *binary document-level classification* problem, where each input text must be labeled as either human-written or machine-generated. Given a complete document, the system is required to assign a single label to the entire text, as illustrated in Fig. 1. To explicitly simulate “in-the-wild” conditions, the evaluation setting departs from standard i.i.d. assumptions. Test documents are sampled from semantic domains that differ from those observed during training and are generated by *undisclosed* large language models. These generators range from vanilla pre-trained models to variants fine-tuned (e.g., via DPO) to better mimic human linguistic distributions. As a result, human- and machine-written documents often discuss highly similar topics, forcing detection systems to rely on subtle stylistic and discourse-level cues rather than topical differences.
- **Subtask B: Human–Machine Text Segmentation.** This subtask is formulated as a *fine-grained boundary localization* problem. Given a document containing both human-written and machine-generated text, the system must identify the exact *character index* at which the machine-generated continuation begins. Each input consists of a variable-length human-authored prompt followed by a model-generated completion. Unlike Subtask A, which requires a single document-level label, this setting demands precise localization within the text, simulating real-world scenarios where MGT is subtly embedded into otherwise authentic content. Fig. 2 shows a representative instance, illustrating how the transition between human and machine text can occur seamlessly within an otherwise coherent document.

A central open question in machine-generated text detection concerns the trade-off between *robustness* and *adaptation*. While supervised models can be fine-tuned to specific languages and domains, their performance may degrade under distribution shift, whereas data-agnostic methods promise greater stability at the cost of accuracy. Understanding this trade-off is particularly critical for under-resourced languages such as Italian, where large-scale native benchmarks are scarce. To investigate this question, we submitted both supervised and unsupervised systems to the DeSegMa-IT shared task, enabling a controlled comparison under identical evaluation conditions.

For Subtask A (document-level detection), we evaluated a zero-shot, domain-agnostic detector based on Fast-DetectGPT [13], a probabilistic method derived from DetectGPT [14] that exploits differences

Example: Subtask A - Machine Generated Text Detection in the Wild

Sample 1: Human-Written (Label 0)

“Viktor Orban, da quando il leghista è diventato ministro dell’Interno, si mandano segnali di apprezzamento reciproco. Un’alleanza che potrebbe portare l’Italia al fianco dei Paesi di Visegrad e che già - in occasione della discussione sulla riforma di Dublino - ha messo in difficoltà gli altri partner dell’Ue. [...]”

Sample 2: Machine-Generated (Label 1)

“Viktor Orban, dopo anni di duri scontri diplomatici, sono pronti a unire le loro forze per riscrivere l’agenda di Bruxelles. Il leader di Fratelli d’Italia Giorgia Meloni e il presidente del governo ungherese si sono visti a Vienna, in un vertice di “centrodestra, identità italiana e sovranità italiana”. [...]”

Goal: The model must classify the entire document binary (0 vs 1). Note the subtle topical similarities making the task challenging.

Figure 1: Instances from the DeSegMa-IT dataset (Subtask A). Both texts discuss similar political topics, requiring the model to detect subtle stylistic or semantic incoherencies rather than simple topic shifts.

Example: Subtask B - Human-Machine Segmentation

Input Document (Mixed Source)

Il presidente del Tribunale internazionale del diritto del mare (Itlos), Vladimir Golitsyn, ha fissato **al 10 agosto la data in cui il tribunale arbitrale di Amburgo esaminerà le informazioni che l’Italia intende raccogliere in India per scagionare i Marò . Nei giorni scorsi su vari quotidiani erano uscite indiscrezioni circa la data di un eventuale incontro...**

- **Human Prompt (Black):** Ends at “...ha fissato ”
- **Machine Continuation (Red):** Starts at “al 10 agosto...”
- **Target Label:** Character Index 103

Figure 2: Instance from Subtask B. The system receives the full text and must predict the character index (103) where the machine-generated continuation begins.

in token-level likelihood distributions between human and machine-generated text. In parallel, we developed supervised classifiers by fine-tuning ModernBERT [15], an English-centric encoder optimized for long-context processing, to assess how effectively cross-lingual transfer can bridge the Italian language gap. For Subtask B (human-machine segmentation), we framed boundary localization as a token-level binary classification problem followed by a change point detection procedure. We compared supervised encoders fine-tuned on the task against an unsupervised, token-level adaptation of Fast-DetectGPT, extending its document-level scoring mechanism to identify transitions within mixed-source texts.

Using the DeSegMa-IT benchmark as a controlled testbed, this work investigates the design space of machine-generated text detection in Italian under realistic, non-I.I.D. conditions. Our study focuses on three main aspects. First, we provide a systematic comparison between data-agnostic, zero-shot detection methods and supervised encoder-based models, analyzing their robustness when both the generating model and the data distribution are unknown. Second, we assess the effectiveness of supervised cross-lingual transfer by fine-tuning English-centric encoders on Italian data and comparing them against unsupervised baselines. Third, we analyze which components of encoder-only architectures contribute most to document-level detection performance through controlled layer-wise ablation experiments.

Overall, our experiments highlight a consistent hierarchy across both tasks: supervised encoder-based models substantially outperform the data-agnostic Fast-DetectGPT baseline in both document-level

detection (Subtask A) and fine-grained segmentation (Subtask B). While supervision proves superior for Italian MGT in the wild, we identify input length as a critical boundary condition, with short texts accounting for a large portion of residual errors. Crucially, through controlled layer-wise ablations, we show that full model depth is not a prerequisite for effectiveness: intermediate encoder layers already encode task-relevant signals that enable competitive detection performance even without relying on final-layer representations.

The remainder of this paper is organized as follows. Section 2 reviews related work, situating our study within the broader landscape of MGT detection. Section 3 describes the submitted systems, detailing both the unsupervised and supervised methodologies. Section 4 presents the experimental results and discussion, including the layer-wise ablation analysis. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related Work

Approaches to machine-generated text (MGT) detection are commonly divided into *supervised* classifiers and *zero-shot* (data-agnostic) methods [13]. Supervised detectors, typically trained on labeled corpora of human and synthetic text, often achieve strong in-domain performance [16], but are known to generalize poorly under distribution shift, particularly when confronted with unseen generators, domains, or decoding strategies [17]. This limitation is especially problematic in real-world scenarios, where the generating model and its alignment properties are unknown [18].

To address robustness, recent benchmarks have emphasized scale and adversarial variation. The RAID dataset [19], comprising over 10 million documents, introduces a wide range of text-level attacks such as homoglyph substitution, whitespace manipulation, and translation-based perturbations. However, a key limitation of such resources is that their synthetic texts are mostly generated using models that lag behind current state-of-the-art open-weight generators (e.g., GPT-2, GPT-3, LLaMA 2, Mistral 7B). As a result, detectors trained on these benchmarks may overfit artifacts that are no longer representative of modern, inexpensive, and highly capable models, raising concerns about the ecological validity of “in-the-wild” settings.

An alternative line of work focuses on *generator-agnostic* signals derived from language model probabilities. DetectGPT [14] exploits the observation that machine-generated text tends to occupy local maxima in the likelihood landscape of models trained on similar data, distinguishing it from perturbed variants. Fast-DetectGPT [13] extends this idea by introducing *conditional probability curvature*, showing that machine-generated text becomes increasingly predictable as generation progresses, while human-authored text remains comparatively flat. These probabilistic approaches exhibit improved robustness across generators, but their sensitivity to decoding parameters, short contexts, increase in temperature parameter and adversarial rewriting remains an open challenge.

Beyond probabilistic cues, recent work highlights *syntactic structure* as a potentially more stable discriminator between human and machine-generated text. Using *Universal Dependencies*, Guo et al. [20] show that LLM-generated text exhibits systematic syntactic divergences from human writing across languages, often retaining an “English accent” even when generating non-English text. For Italian in particular, prior studies report that model performance degrades with increasing syntactic complexity, such as inversions or post-verbal subjects [21]. Given Italian’s rich morphology and relatively flexible word order, such deviations may provide discriminative signals that are less sensitive to surface-level paraphrasing. These observations suggest that intermediate syntactic representations, rather than surface fluency alone, may offer more stable cues for machine-generated text detection under paraphrasing or stylistic manipulation.

These issues are amplified in lower-resourced languages, where both training data and evaluation benchmarks are scarce [22, 23]. For Italian, the only publicly documented evaluation of MGT detection to date is the multilingual setting of Task 8 at SemEval 2024 [8], where Italian appeared as a *surprise language*. That evaluation relied on a relatively small dataset (approximately 3,000 human and 3,000 machine-generated texts), with synthetic data produced by a fine-tuned LLaMA 2 70B model [24],

and did not explicitly model adversarial conditions or generator diversity. No large-scale adversarial benchmark comparable to RAID currently exists for Italian, and most available resources focus on general NLP evaluation rather than robustness in machine-text detection [25, 26]. As a result, the current empirical understanding of MGT detection for Italian remains limited to small-scale, non-adversarial settings.

In this context, data-agnostic approaches are particularly attractive. Prior work suggests that some probabilistic detection signals transfer across languages: DetectGPT scores have been shown to correlate across translations of the same text, including Italian [27, 14]. At the same time, the evidence on syntactic divergence suggests that language-specific structural cues may be crucial for reliable detection in morphologically rich languages.

Building on these insights, our work combines zero-shot probabilistic detectors with supervised encoder-based models and layer-wise analysis. We focus on ModernBERT [15], whose alternation of local and global attention layers is well-suited to capturing both document-level coherence and intermediate syntactic representations. By jointly evaluating data-agnostic and supervised approaches on the same benchmark and analyzing which internal representations drive performance, we aim to clarify not only *whether* MGT detection works for Italian, but *which signals* are effective, *where they emerge within encoder representations*, and *under which conditions* they remain reliable.

3. Description of the Systems

This section describes the systems submitted to the DeSegMa-IT shared task. We outline the modeling assumptions and architectural choices adopted for each subtask, contrasting data-agnostic and supervised approaches under identical evaluation conditions. Beyond system description, this section is structured to highlight how specific design factors, such as context granularity, representation depth, and training data diversity, shape robustness and performance in machine-generated text detection.

3.1. Subtask A: Human–Machine Text Detection

Subtask A requires binary document-level classification of human-written versus machine-generated text under realistic, non-I.I.D. conditions. Rather than assuming access to the generating model or domain, the task challenges systems to operate under a distribution shift. Accordingly, we approach detection from two complementary perspectives: a data-agnostic, zero-shot baseline and supervised encoder-based models trained to adapt to Italian data.

3.1.1. Unsupervised Setting: Fast-DetectGPT

For Subtask A, we use Fast-DetectGPT [13] as a data-agnostic baseline. Fast-DetectGPT is a zero-shot detector that relies on token-level probability estimates produced by a language model to distinguish human-authored from machine-generated text.

The core idea is that machine-generated text tends to follow the probability distribution of any language model more consistently than human text. Given a text $x = (x_1, \dots, x_T)$ and a scoring model p_θ , the method analyzes each token in context. At position t , it compares the log-probability of the observed token, $\log p_\theta(x_t \mid x_{<t})$, with the expected log-probability under the same predictive distribution, corresponding to the negative entropy of $p_\theta(\cdot \mid x_{<t})$. This yields a local discrepancy score indicating whether the token is more predictable than an average continuation according to the model.

Since this computation is performed at the token level, Fast-DetectGPT produces a sequence of discrepancy scores. To obtain a document-level decision, these scores are aggregated across the sequence and normalized for length and variance, yielding a single scalar score $d(x)$. Texts that consistently align with the model’s expectations produce positive normalized scores, while human-authored texts tend to fluctuate around zero. Final classification is obtained by thresholding $d(x)$.

Token probabilities, and thus the detection signal, depend on the choice of the scoring language model. Using a model close to the generator strengthens this signal. Accordingly, Fast-DetectGPT

distinguishes between a *white-box* setting, where the scoring model matches or closely approximates the generator, and a *black-box* setting, where a surrogate model is used. Here, *white-box* does not imply access to model parameters, but rather membership in the same model family.

In our experiments, we evaluated multiple scoring models (Llama-3.1-8B-Instruct, Gemma-3-12B-IT, Falcon-7B-Instruct, BLOOM-7B1) on the training split. Llama-3.1-8B-Instruct achieved the highest accuracy (0.948) and was selected for the final submission as a quasi-white-box configuration, since the generators in the test set are undisclosed.

3.1.2. Supervised Setting: Encoder-Based Classification

Since Subtask A is formulated as a document-level binary classification problem, we adopt a supervised encoder-based approach. Transformer encoders are a natural choice in this setting: self-attention enables the integration of evidence across long contexts, while internal representations can encode syntactic and structural regularities that differ between human- and machine-generated text. This perspective is particularly relevant for Italian, where syntactic variation, flexible word order, and rich morphosyntactic agreement provide potential cues beyond surface fluency. Recent work suggests that LLM-generated text, especially when produced by English-centric models, may retain subtle structural biases that manifest at the syntactic level. Accordingly, we investigate whether detection signals are better captured through representations that emphasize structure rather than purely semantic abstraction.

A central modeling question therefore concerns the granularity at which such discriminative signals emerge. Some artifacts of machine-generated text may be highly localized (e.g., repetitive phrasing or short-range regularities), while others may only become apparent when considering global discourse organization and coherence. To study this trade-off, we explore two complementary training regimes. In a *local-context* setting, documents are split into overlapping chunks of three sentences, which are classified independently. This increases the number of training instances and emphasizes short-range artifacts. At inference time, chunk-level predictions are aggregated via majority voting to obtain a document-level label. In a *global-context* setting, the encoder processes the entire document, allowing the model to exploit long-range discourse structure. For full-document training, inputs are truncated to 1,024 tokens, which covers the majority of documents in our data.

Within the chunk-based regime, we further examine where discriminative information is encoded inside the model. BERT-like encoders distribute linguistic information across layers: intermediate representations often make syntactic structure more linearly accessible than the top layer [28]. We therefore evaluate features extracted not only from the final [CLS] embedding, but also from intermediate layers. This enables a controlled assessment of whether syntactic representations contribute to MGT detection. These design choices directly motivate the layer-wise ablation analysis reported in Section 4 and are grounded in the probing results presented in Appendix A.

All supervised systems follow a standard encoder-classifier architecture. A pretrained transformer encoder processes the input text, and a lightweight classification head predicts a single document-level label. Our primary backbone is ModernBERT [15], an English-centric encoder optimized for long-context processing through an alternation of local and global attention layers. We additionally consider mModernBERT [29] as comparative baseline.

Finally, we investigate the effect of model capacity and pretraining. We compare ModernBERT-base and ModernBERT-large to assess whether increased capacity improves cross-lingual transfer, and contrast English-centric and multilingual encoders under the same training protocol. All supervised models are fine-tuned using cross-entropy loss.

For the official submission, we selected ModernBERT-large trained on an augmented dataset with full-document context, prioritizing robustness to unseen generators. A post-hoc analysis, discussed in Section 4.1, shows that the same architecture trained exclusively on the original dataset achieves the highest accuracy overall.

3.2. Subtask B: Human - Machine Text Segmentation

Subtask B addresses a more fine-grained and realistic scenario in which human-written and machine-generated text are interwoven within the same document. Given a text that begins with a human-authored prompt and continues with a machine-generated segment, the goal is to identify the character index at which the transition occurs. Unlike Subtask A, which focuses on document-level classification, this setting requires precise boundary localization. As in Subtask A, we investigate both a data-agnostic approach and a supervised method to analyze their effectiveness under these conditions.

3.2.1. Unsupervised Setting: Token-Level Fast-DetectGPT Adaptation

As a data-agnostic baseline for Subtask B, we reuse Fast-DetectGPT [13], introduced in Section 3.1.1, and investigate whether its detection signal can be repurposed for boundary localization. All experiments employ the same scoring model selected for Subtask A, namely Llama-3.2-8B-Instruct, to ensure consistency across tasks. In Subtask A, Fast-DetectGPT aggregates token-level discrepancy scores into a single document-level statistic to determine whether a text is human- or machine-generated. In Subtask B, however, each document is composed of both human-authored and machine-generated segments. This setting makes a global aggregation unsuitable, as the detection signal is expected to vary within the same document. A natural extension is therefore to inspect the Fast-DetectGPT discrepancy signal locally at the token level.

Our intuition is that mixed human-machine text should exhibit two distinct statistical regimes: an initial human-authored region characterized by near-zero discrepancy scores, followed by a machine-generated region with consistently positive values. While we cannot use those individual values to predict the exact splitting point, we hypothesize that we can use those aggregated scores to find an area in which the slope of the cumulative discrepancy score sharply increases, and use that as a boundary detection estimator.

To make this intuition operational, we retain the original Fast-DetectGPT discrepancy score s_j formulation. Instead of aggregating only at the end of the paragraph, for each token position j we compute the relative cumulative discrepancy score:

$$s_j = \log P(x_j \mid x_{<j}) + H_j \quad (1)$$

$$H_j = - \sum_{v \in \mathcal{V}} P(v \mid x_{<j}) \log P(v \mid x_{<j}) \quad (2)$$

Where $\log P(x_j \mid x_{<j})$ is the log-probability of the observed token given the preceding context, and H_j is the entropy of the predictive distribution at position j . Intuitively, s_j measures how much an observed token is more likely than a randomly sampled alternative: machine-generated tokens tend to occupy high-probability regions, yielding positive scores, while human-authored tokens, even cumulated, center around ≈ 0 [14].

Since boundary localization requires stability over noise, we aggregate these scores cumulatively using a normalized cumulative statistic in which where σ_i^2 is the variance of $\log P(v \mid x_{<i})$ under the model’s predictive distribution:

$$Z_j = \frac{\sum_{i=0}^j s_i}{\sqrt{\sum_{i=0}^j \sigma_i^2}} \quad (3)$$

We rely on heuristic strategies to infer the transition point from the normalized cumulative trajectory Z_j . Since we are looking at the point in which the slope changes, we propose a combined score method that leverages both the first derivative (slope) and second derivative (curvature) of the smoothed cumulative trajectory Z_j . The trajectory is first smoothed using a Gaussian filter to reduce noise, after which both derivatives are computed and independently normalized to the $[0, 1]$ range to ensure comparability. The final decision signal is an average of the normalised first and second derivatives.

The predicted boundary is the position maximizing this combined signal, excluding a margin of 10% at both ends to avoid spurious detections near document boundaries, respecting the peculiarity of the challenge data set, in which the tails are always human.

3.2.2. Supervised Setting: Token-Level Classification

For the supervised setting of Subtask B, we address boundary detection by directly learning token-level cues from data. While the unsupervised approach relies on word-level predictability estimated by a language model, here we ask whether similar information can be learned implicitly by an encoder trained on labeled examples.

We fine-tune a transformer encoder to perform binary token-level classification, where each token is labeled as either human-authored (0) or machine-generated (1). The intuition mirrors the unsupervised case: machine-generated continuations tend to be locally more regular and predictable, while human text exhibits higher variability. However, predicting whether a single token corresponds to the exact transition point is an extremely challenging objective. As discussed in the unsupervised setting, individual tokens are often highly constrained by syntax and discourse, regardless of authorship.

Crucially, the output of the encoder is not a hard decision but a probability distribution. For each token t_i , the model produces a probability $p_i \in [0, 1]$ indicating the likelihood that the token belongs to the machine-generated segment. As in BERT-style encoders, these probabilities are context-dependent and reflect information aggregated over the surrounding tokens. While the classifier is not expected to be reliable exactly at the boundary, it can reliably identify regions that are clearly human-written or clearly machine-generated.

For this reason, boundary detection is not performed by thresholding individual token predictions. Moreover, the heuristics used in the unsupervised setting are not applicable here, as they are defined over word-level probability distributions rather than learned classification scores. Instead, we frame boundary localization as a change point detection problem over the sequence of predicted probabilities.

Given a sequence of N tokens with associated probabilities $\{p_1, \dots, p_N\}$, we score each candidate boundary position k using the following change point detection formula:

$$\text{Score}(k) = \sum_{i=1}^k \log(1 - p_i) + \sum_{j=k+1}^N \log(p_j) \quad (4)$$

The first term accumulates evidence that tokens before k are human-authored, while the second term accumulates evidence that tokens after k are machine-generated. The predicted boundary is selected as:

$$k^* = \arg \max_k \text{Score}(k)$$

This formulation favors split points where confident human predictions are concentrated on the left and confident machine predictions on the right, while uncertain tokens near the boundary have limited influence. We employ ModernBERT as the primary encoder, augmented with a two-layer MLP classification head. As a follow-up analysis, we replicate the same training procedure using mModernBERT to assess the impact of multilingual pretraining under limited Italian supervision. Token indices are finally mapped to character offsets by selecting the first character of the predicted boundary token.

4. Results and Discussion

This section details the experimental evaluation of the systems described in the Section 3. We structure the analysis by subtask, first defining the experimental protocol and baselines, and then discussing the results with a distinction between the official submission and post-hoc analyses. Finally, we synthesize our findings regarding the broader research questions on robustness and representation depth.

4.1. Subtask A: Machine-Generated Text Detection

In this section, we report results on the official blind test set of the shared task, in which both domains and generators are unseen during training. We compare supervised encoders with zero-shot baselines, analyze the effects of data augmentation and context size, and discuss which architectural choices support effective cross-lingual detection.

4.1.1. Experimental Setup

Data Distribution. Experiments were conducted on the official DeSegMa-IT training dataset. While the class distribution is perfectly balanced (50% human, 50% machine), we observe a significant *length asymmetry*: human-authored texts are on average twice as long as machine-generated ones (799 vs 406 tokens) and exhibit much higher variance. This “brevity bias” acts as a potential confounder, as a model might learn to associate shortness with artificiality. To improve robustness against unseen generators, we adopted an augmented training regime by injecting 14,000 additional instances: 7,000 human articles from the CHANGE-IT² corpus of Italian newspaper articles and 7,000 synthetic samples generated by translating the English RAID benchmark [19]³ via Qwen2.5-7B-Instruct.

The blind test-set of the is composed by 15,135 instances.

Investigated Systems. To disentangle the contribution of architecture, context size, and training data, we benchmark a comprehensive set of configurations corresponding to the entries in Table 1:

- **Unsupervised Baseline (Fast-DetectGPT):** We employ Fast-DetectGPT with Llama-3.2-8B-Instruct as the scoring model. This serves as a data-agnostic reference point, quantifying how much detection signal is available purely from distributional artifacts (log-likelihood curvature) without access to labeled training examples.
- **Supervised Backbone (ModernBERT):** Our primary analysis relies on ModernBERT-large, evaluated under three distinct regimes to test robustness:
 1. *Local:* The model classifies independent 3-sentence windows, aggregating predictions via majority vote⁴. This tests detection reliability in low-context scenarios.
 2. *Global:* The model processes the entire document (up to 8,192 tokens) to capture long-range discourse incoherence. This represents our strongest configuration trained on the official data.
 3. *Global + Aug (Submitted):* The official submission configuration, identical to the Global setting but trained on the augmented dataset (including RAID Qwen-translated texts) to test generalization to unseen generators.
- **Comparative Architectures:** To validate whether ModernBERT’s performance stems from its architecture or cross-lingual transfer, we compare it against **mModernBERT**: the multilingual variant of our primary backbone. Comparing this against the English-centric ModernBERT allows us to isolate the impact of language-specific pretraining versus structural transfer.
- **Most Frequent Class (MFC):** A trivial baseline ($\approx 50.10\%$) included solely to establish the random-guess floor for the balanced test set.

The selection of the submitted configuration (ModernBERT Large with Data Augmentation) was necessitated by the lack of a discriminative signal on our internal validation set. During the development phase, we observed a pronounced ceiling effect, where several explored configurations achieved near-perfect performance, rendering them empirically indistinguishable. Consequently, we prioritized the system trained with Data Augmentation based on the hypothesis that exposure to a more diverse data

²<https://huggingface.co/datasets/MattiaSangermano/change-it>

³<https://raid-bench.xyz/>

⁴Technically, we employ a *soft-voting* aggregation strategy. Rather than counting discrete class labels, we extract the softmax probability $P(\text{Machine} \mid c_i)$ vs. $P(\text{Human} \mid c_i)$ and use them for weighted voting.

distribution would bring superior generalization capabilities and robustness against distribution shifts in the blind test set.

Metrics. Primary evaluation is based on Accuracy, the official metric for Subtask A. Additionally, we report Precision, Recall, and Macro-F1 to provide a granular view of error types and class-specific behavior.

4.1.2. Main Results

Table 1 presents the results on the official test set. The table shows our unsupervised model based on Fast-DetectGPT and the naive baseline in the first two rows. Inside rows 3-5 we display our supervised models based on the ModernBERT architecture. On row 6 we present the best competitor system for reference.

The hierarchy of performance provides clear answers to our research questions regarding the trade-off between robustness and adaptation.

Supervised vs. Unsupervised. The first major finding is the substantial gap between data-agnostic and supervised methods. Fast-DetectGPT achieves a respectable accuracy of 82.86%, confirming that machine-generated texts in Italian leave detectable statistical traces (e.g., negative curvature regions in log-likelihood) that persist even zero-shot. However, supervised adaptation yields a relative error reduction of over 75% (best supervised model: 95.78%). This suggests that while generic statistical artifacts are present, the specific generators used in the wild exhibit idiosyncratic patterns that are best captured through fine-tuning.

The Failure of Naive Augmentation. A counter-intuitive result concerns our submission strategy. Our official run, trained on the *Augmented* dataset (which included translated RAID data and Qwen-generated texts), reached 92.87% accuracy. However, our post-hoc analysis reveals that training *exclusively* on the official provided data (Original) would have yielded significantly higher performance (95.78%), outperforming the best competitor system (94.57%). This indicates that the augmentation strategy introduced a distributional shift that was detrimental rather than helpful. The specific stylistic signatures of the "in-the-wild" test set were likely well-represented in the original training data. By injecting synthetic data from different domains and generators (e.g., Qwen), we likely diluted the model's focus, validating the hypothesis that MGT detection is highly sensitive to the specific generator-domain pairs used for training.

Context Matters: Local Context vs. Global Context. We compare processing granularity by contrasting a chunk-based approach (voting on 3-sentence windows) against a full-document approach. The full-document model outperforms the chunked variant by nearly 10 percentage points (95.78% vs 85.97%). This result has a crucial theoretical implication: detection signals are not merely local glitches (e.g., repetition or non-sequiturs visible in a short window) but are deeply rooted in global discourse coherence. The model requires long-range attention to detect the subtle "fading" of logic or structure typical of neural generation.

Architectural Analysis: Cross-Lingual Transfer. A striking observation from Table 1 is the performance of the English-centric ModernBERT-large (95.78%) compared to its multilingual counterpart mModernBERT (92.82%). Conventional wisdom suggests that for Italian tasks, a multilingual model should be superior. Why does an English-centric model win? We argue that, at document-level, MGT detection relies less on language-specific lexical knowledge (vocabulary coverage) and more on *structural* and *distributional* features (e.g., entropy plateaus, syntactic regularity) which are artifacts of the Transformer architecture itself. Since LLMs (the generators) share similar underlying architectures regardless of the output language, the artifacts they introduce are likely universal. ModernBERT's

Table 1

Subtask A results on the **official test set**. All values are reported in percentages (%). **Submitted** denotes our official competition entry. The *Best Competitor System* corresponds to the first-place submission by team *Gradient Descenders*. *MFC* denotes the Most Frequent Class baseline.

Run	Model	Acc	P	R	F1
Unsupervised	MFC (Most Frequent Class Baseline)	50.10	25.05	50.00	33.38
	Fast-DetectGPT (scorer: LLama3 . 2-8B)	82.86	84.68	82.88	82.64
Supervised	ModernBERT-large (local)	85.97	89.02	85.97	87.47
	mModernBERT (global)	92.82	93.66	92.78	92.78
	ModernBERT-large (global + aug) (Submitted)	92.87	93.63	92.85	92.82
	ModernBERT-large (global)	95.78	95.99	95.78	95.77
NA	Best Competitor System	94.57	-	-	-

superior pre-training on code and massive English corpora likely allowed it to learn more robust representations of "artificiality" and structural coherence, which transferred zero-shot to Italian syntax more effectively than the noisy multilingual alignment of mBERT variants.

The "Safe Zone" of Detection. Table 2 breaks down performance by document length. The results highlight a clear boundary condition for encoder-based detectors. On documents longer than 512 tokens, the model is effectively infallible (99.9% accuracy). This confirms that given sufficient statistical evidence, the distinction between human and machine text is sharp.

The Short-Text Failure Mode. Conversely, performance drops drastically to 76.8% for texts shorter than 128 tokens. This correlates with the "Brevity Bias" observed in the training data: the model struggles to disentangle "shortness" from "artificiality," or simply lacks the necessary context window to observe discourse-level incoherence. This remains the primary open challenge for future research: detecting MGT in low-resource (short context) regimes requires signal amplification strategies that standard fine-tuning does not provide.

Table 2

Subtask A: Error Analysis by Document Length (ModernBERT-large). The number of tokens is calculated based on the ModernBERT tokenizer.

Text Length (tokens)	Accuracy(%)
Short (< 128)	76.83
Medium (128-512)	89.95
Long (> 512)	99.90

Layer-Wise Ablation Study. To determine whether MGT detection relies primarily on low-level structural artifacts or requires high-level semantic abstraction, we conducted a layer-wise ablation study using ModernBERT-base in the local-context (chunked) setting. Results are detailed in Table 4 in appendix A. Contrary to the hypothesis that detection signals saturate in intermediate syntactic layers, we observe a monotonic improvement in performance throughout the network depth. While the first layer already provides a strong baseline (Accuracy: 81.23%), indicating the presence of surface-level lexical or morphological cues, reliance on intermediate representations (Layers 11-14) yields only partial gains (84.63%). The highest performance is achieved strictly at the final classification layer (87.15%). This trajectory suggests that while "glitches" in machine-generated text may begin at the surface level, reliable detection of advanced generators requires the full compositional capacity of the Transformer to model complex discourse-level dependencies.

4.2. Subtask B: Human–Machine Text Segmentation

4.2.1. Experimental Setup

Data Distribution. Experiments focus on the training and test split of the DeSegMa-IT Subtask B dataset, comprising 19,945 mixed-source documents for the training section and of 23,211 documents for the test section.

Each instance consists of a human-authored prompt followed by a machine-generated continuation. An analysis of the training distribution reveals a critical structural property: the transition point is located, on average, at character 264 (± 116.9). This implies that the human context is relatively short (averaging 84.5 tokens), while the machine-generated continuation is dominant (averaging 355.0 tokens). This asymmetry poses a significant modeling challenge: the encoder must detect the stylistic shift relying on a very limited "warm-up" window of human text to establish a baseline. Finally, the total document length in the training set remains well within the capacity of ModernBERT’s context window, ensuring that boundary artifacts are not conflated with truncation noise during fine-tuning.

Investigated Systems. To isolate the contributions of supervision and multilingual pretraining to boundary localization, we benchmark the following configurations corresponding to the entries in Table 3:

- **Unsupervised Baseline (Token-Level Fast-DetectGPT):** We adapt Fast-DetectGPT to the segmentation setting by retaining its token-level discrepancy scores rather than aggregating them into a single document statistic. Using Llama-3.1-8B-Instruct as the scoring model (consistent with Subtask A), we compute normalized cumulative trajectories and apply a combined first- and second-derivative heuristic to detect the point of maximum slope change, as described in Section 3.1.1. This serves as a data-agnostic reference point, testing whether local probability curvature alone can signal authorship transitions.
- **Supervised Backbone (ModernBERT-large, Submitted):** Our official submission employs ModernBERT-large fine-tuned for token-level binary classification, where each token is labeled as human-authored or machine-generated. A two-layer MLP classification head produces per-token probabilities, and final boundary prediction is obtained via the change point detection procedure described in Equation (4). Token indices are mapped to character offsets by selecting the first character of the predicted boundary token.
- **Multilingual Variant (mModernBERT):** To assess whether fine-grained segmentation benefits from native morphological representations, we replicate the supervised training procedure using mModernBERT-base. This comparison isolates the impact of multilingual pretraining versus cross-lingual transfer from an English-centric encoder.
- **Middle Split Baseline:** A naive baseline that places the boundary at 50% of the document length, included to establish the difficulty floor for the localization task.

Metrics Evaluation is performed using Mean Absolute Error (MAE), which measures the average character distance between the predicted boundary and the ground truth. Unlike binary classification metrics, MAE quantifies the precision of localization, penalizing errors proportionally to their magnitude. To benchmark the difficulty of the task, we introduce a naive Middle-Split Baseline, which blindly places the boundary at 50% of the document length.

4.2.2. Main Results

Table 3 summarizes Subtask B performance measured by Mean Absolute Error (MAE) in characters. In the first two rows is presented a naive splitting at the 50% mark to use as reference and our unsupervised Token-level Fast-DetectGPT. On rows 3-4 we display our supervised models based on the ModernBERT architecture employing the change point detection formula 3.2.2. On row 5 we present the best system presented in the challenge as a point of reference.

Our comparative analysis targets two specific dimensions of the segmentation problem. First, we assess the viability of Zero-Shot Curvature Estimation by applying Token-Level Fast-DetectGPT, testing whether local perplexity spikes are sufficient to signal authorship changes without supervision. The token-level Fast-DetectGPT adaptation achieves 158.31 MAE, a substantial improvement over the middle-split baseline (695.37) but nearly double the error of our submitted supervised system. This gap suggests that local probability curvature, while informative at the document level, provides insufficient granularity for precise boundary localization without a different heuristic compared to the usage of derivatives or additional syntactic informations.

Second, within the supervised framework, we investigate the role of Multilingual Alignment. By comparing the English-centric ModernBERT (our official submission) against the multilingual mModernBERT (post-hoc), testing the hypothesis that fine-grained segmentation, unlike document-level detection, requires the native morphological representations offered by multilingual pretraining.

Our submitted ModernBERT-large system achieves 81.6 MAE, corresponding to an average localization error of approximately 13 words, in line with our a-priori evaluation on an held-out portion of the training set. The post-hoc experiments with mModernBERT yield a 25.7% relative improvement, reducing MAE to 60.58 characters and narrowing the gap with the best competitor to approximately 8 characters. This result contrasts with Subtask A, where ModernBERT-large outperformed MultilingualModernBERT-base by almost 3%. We attribute this reversal to the differing task demands: token-level boundary detection requires fine-grained morphosyntactic sensitivity that multilingual pre-training better provides, whereas document-level classification benefits from global coherence signals that transfer well cross-lingually.

Table 3

Subtask B results on the **official test** (MAE in characters; lower is better). **Submitted** indicates the official run.

Run	Model	MAE
Unsupervised	Middle Split Baseline	695.37
	Token-level Fast-DetectGPT (scorer: LLama3 . 2 - 8B)	158.31
Supervised	ModernBERT + Change-Point (Submitted)	81.60
	mModernBERT + Change-Point	60.58
NA	Best Competitor System	52.54

5. Conclusions

This work compared supervised and unsupervised approaches for Italian machine-generated text detection on the DeSegMa-IT benchmark. For Subtask A (document-level classification), our official submission, ModernBERT-large with data augmentation and full-document context, achieved 92.87% accuracy, placing third in the challenge. The same ModernBERT-large trained without data-augmentation for three epochs achieved 95.78% accuracy in a successive ablation study, while MultilingualModernBERT-base scored 92.82%. Layer-wise ablations on ModernBERT-base using chunk-based classification reveal that intermediate layers (11–14) achieve 84.6% accuracy compared to 87.2% for the final [CLS] embedding, suggesting that syntactic representations in the encoder’s middle layers already capture most of the discriminative signal. For Subtask B (boundary detection), the supervised ModernBERT-large system with change point detection obtained 81.6 MAE, which improved to 60.6 MAE when replaced with the multilingually pre-trained MultilingualModernBERT-base. This reversal shows that while *model size* is the decisive factor for Subtask A, *multilingual pretraining* is more relevant in the context of Subtask B. This confirms our hypothesis that MGT detection at the document level relies more on structural features that generalize across languages, while fine-grained segmentation requires native semantic and morphological knowledge.

Our submitted model scored almost perfectly - 99.9%+ accuracy - on texts longer than 512 tokens, dropping to 76.83% for short texts under 128 tokens, with almost all errors being false negatives. If

those findings were replicated consistently on different text-domains and against adversarial prompting and ablated models, automatic MGT detectors can be suggested for usage on long texts.

For both experiments the unsupervised domain-agnostic systems, predictably, scored behind their supervised counterparts. On Subtask A, Fast-DetectGPT using LLama3.2-8B as evaluator reached 84%, while the token-level Fast-DetectGPT adaptation yielded 158 MAE on Subtask B, suggesting that local probability curvature alone is insufficient for precise boundary localization. Contrary to our expectation, adding 14,000 new documents coming from RAID [19] as Data Augmentation for Subtask A reduced by almost 3% the model performance. This suggests that augmentation with translated data introduces distributional artifacts that outweigh robustness gains on this benchmark. Since FastDetectGPT makes no prior assumptions about the language or domain, we expect its performance to remain flat across domains, registers, and languages, while supervised model performance should decrease with respect to distance to the training dataset. This makes Fast-DetectGPT, and unsupervised methods in general, paradoxically more suited to real-world application than benchmarks. Prior work has shown that multilingual LLMs retain English-influenced syntactic patterns even when generating in other languages [20], and our probing analysis (Appendix A) reveals that hierarchical depth information remains weakly encoded in ModernBERT. We believe that incorporating explicit syntactic features alongside semantic signals offers currently the most promising avenue for exploration.

Acknowledgments

The authors acknowledge financial support from the PNRR MUR project PE0000013-FAIR. Moreover, the authors acknowledge support from Project ECS 0000024 Rome Technopole, - CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, Funded by the European Union - NextGenerationEU.

Declaration on Generative AI

Parts of the writing and editing process for this manuscript were supported by the use of generative AI tools, specifically Google’s Gemini. These tools were employed to enhance clarity and correct spelling. All AI-assisted content was carefully reviewed and validated by the authors to ensure accuracy, originality, and compliance with ethical and scientific standards. The authors bear full responsibility for the final content.

References

- [1] A. Liu, A. Mei, B. Lin, B. Xue, B. Wang, B. Xu, B. Wu, B. Zhang, C. Lin, C. Dong, et al., Deepseek-v3. 2: Pushing the frontier of open large language models, arXiv preprint arXiv:2512.02556 (2025).
- [2] K. Team, Y. Bai, Y. Bao, G. Chen, J. Chen, N. Chen, R. Chen, Y. Chen, Y. Chen, Y. Chen, et al., Kimi k2: Open agentic intelligence, arXiv preprint arXiv:2507.20534 (2025).
- [3] A. Arditì, O. Obeso, A. Syed, D. Paleka, N. Panickssery, W. Gurnee, N. Nanda, Refusal in language models is mediated by a single direction (2024). arXiv:2406.11717.
- [4] A. K. Kadhim, L. Jiao, R. A. Shafik, O.-C. Granmo, Adversarial attacks on ai-generated text detection models: A token probability-based approach using embeddings, 2025. URL: <https://arxiv.org/abs/2501.18998>. arXiv:2501.18998, preprint.
- [5] Y. Zhou, B. He, L. Sun, Humanizing machine-generated content: Evading AI-text detection through adversarial attack, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 8427–8437. URL: <https://aclanthology.org/2024.lrec-main.739/>.
- [6] A. Uchendu, Z. Ma, T. Le, R. Zhang, D. Lee, TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for

- Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 2001–2016. URL: <https://aclanthology.org/2021.findings-emnlp.172/>. doi:10.18653/v1/2021.findings-emnlp.172.
- [7] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is chatgpt to human experts? comparison corpus, evaluation, and detection, 2023. URL: <https://arxiv.org/abs/2301.07597>. arXiv:2301.07597.
 - [8] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. Mohammed Afzal, T. Mahmoud, G. Puccetti, T. Arnold, SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection, in: A. K. Ojha, A. S. Doğruöz, H. Tayyar Madabushi, G. Da San Martino, S. Rosenthal, A. Rosá (Eds.), Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 2057–2079. URL: <https://aclanthology.org/2024.semeval-1.279/>. doi:10.18653/v1/2024.semeval-1.279.
 - [9] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, C. Whitehouse, O. M. Afzal, T. Mahmoud, T. Sasaki, et al., M4: Multi-generator, multi-domain, and multi-lingual black-box machine-generated text detection, in: Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 1369–1407.
 - [10] G. Sarti, M. Nissim, It5: Text-to-text pretraining for italian language understanding and generation, in: International Conference on Language Resources and Evaluation, 2022. URL: <https://api.semanticscholar.org/CorpusID:247315276>.
 - [11] G. Puccetti, A. Pedrotti, A. Esuli, Desegma-it at evalita 2026: Overview of the detection and segmentation of machine generated text in italian task, 2026.
 - [12] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [13] G. Bao, Y. Zhao, Z. Teng, L. Yang, Y. Zhang, Fast-detectgpt: Efficient zero-shot detection of machine-generated text via conditional probability curvature, in: Proc. of ICLR, 2023.
 - [14] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, C. Finn, Detectgpt: zero-shot machine-generated text detection using probability curvature, in: Proceedings of the 40th International Conference on Machine Learning, ICML’23, JMLR.org, 2023.
 - [15] B. Warner, A. Chaffin, et al., Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 2526–2547. URL: <https://aclanthology.org/2025.acl-long.127/>. doi:10.18653/v1/2025.acl-long.127.
 - [16] S. T. Lekkala, Y. Annepaka, A. K. Challa, S. R. Machireddy, P. Pakray, C. Chunka, Mixture of detectors: A compact view of machine-generated text detection, arXiv preprint arXiv:2509.22147 (2025).
 - [17] J. Pu, Z. Sarwar, S. M. Abdullah, A. Rehman, Y. Kim, P. Bhattacharya, M. Javed, B. Viswanath, Deepfake text detection: Limitations and opportunities, in: 2023 IEEE Symposium on Security and Privacy (SP), 2023, pp. 1613–1630. doi:10.1109/SP46215.2023.10179387.
 - [18] F. Borazio, C. D. Hromei, E. Passone, D. Croce, R. Basili, Mm-iglu-it: Multi-modal interactive grounded language understanding in italian, in: A. Artale, G. Cortellessa, M. Montali (Eds.), AIXIA 2024 – Advances in Artificial Intelligence, Springer Nature Switzerland, Cham, 2025, pp. 64–78.
 - [19] L. Dugan, A. Hwang, F. Trhlík, A. Zhu, J. M. Ludan, H. Xu, D. Ippolito, C. Callison-Burch, RAID: A shared benchmark for robust evaluation of machine-generated text detectors, in: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 12463–12492. URL: <https://aclanthology.org/2024.acl-long.674>.
 - [20] Y. Guo, S. Conia, Z. Zhou, M. Li, S. Potdar, H. Xiao, Do large language models have an english accent? evaluating and improving the naturalness of multilingual llms, in: Proceedings of the

- 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 3823–3838.
- [21] E. S. Ruzzetti, F. Ranaldi, D. Onorati, D. Venditti, L. Ranaldi, T. Caselli, F. M. Zanzotto, Assessing the asymmetric behaviour of italian large language models across different syntactic structures, in: Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), 2024, pp. 854–863.
 - [22] D. Macko, J. Kopal, R. Moro, I. Srba, Multisocial: Multilingual benchmark of machine-generated text detection of social-media texts, in: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 727–752.
 - [23] O. Al Minshidawi, A.-H. Vahabie, Classifying ai-generated text in low-resource languages like arabic, *AUT Journal of Modeling and Simulation* 57 (2025) 113–124.
 - [24] Y. Wang, J. Mansurov, P. Ivanov, J. Su, A. Shelmanov, A. Tsvigun, O. Mohammed Afzal, T. Mahmoud, G. Puccetti, T. Arnold, A. Aji, N. Habash, I. Gurevych, P. Nakov, M4GT-bench: Evaluation benchmark for black-box machine-generated text detection, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 3964–3992. URL: <https://aclanthology.org/2024.acl-long.218/>. doi:10.18653/v1/2024.acl-long.218.
 - [25] B. Magnini, R. Zanoli, M. Resta, M. Cimmino, P. Albano, M. Madeddu, V. Patti, Evalita-LLM: Benchmarking large language models on italian, *arXiv preprint arXiv:2502.02289* (2025). [arXiv:2502.02289](https://arxiv.org/abs/2502.02289).
 - [26] L. Moroni, A. Esuli, D. Marcheggiani, B. Magnini, Towards a more comprehensive evaluation for italian LLMs, in: Proceedings of the 2024 Workshop on Computational Linguistics for Italian (CLiCit 2024), 2024, pp. 48–59.
 - [27] A. Esuli, F. Falchi, M. Malvaldi, G. Puccetti, You write like a GPT, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 343–348. URL: <https://aclanthology.org/2024.clicit-1.41/>.
 - [28] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4129–4138. URL: <https://aclanthology.org/N19-1419/>. doi:10.18653/v1/N19-1419.
 - [29] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, B. Van Durme, mmbert: A modern multilingual encoder with annealed language learning, *arXiv preprint arXiv:2509.06888* (2025). URL: <https://arxiv.org/abs/2509.06888>.
 - [30] L. Schut, Y. Gal, S. Farquhar, Do multilingual llms think in english?, in: ICLR 2025 Workshop on Building Trust in Language Models and Applications, 2025.
 - [31] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, M. Baroni, What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 2126–2136. URL: <https://aclanthology.org/P18-1198>. doi:10.18653/v1/P18-1198.
 - [32] J. Hewitt, P. Liang, Designing and interpreting probes with control tasks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2733–2743. URL: <https://aclanthology.org/D19-1275>. doi:10.18653/v1/D19-1275.
 - [33] G. Jawahar, B. Sagot, D. Seddah, What does BERT learn about the structure of language?, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 3651–3657. URL: <https://aclanthology.org/P19-1356/>. doi:10.18653/v1/P19-1356.

A. Layer Ablation and Probing Analysis

We provide here the grounding and detailed analysis for the layer-wise ablation study briefly discussed in Section 4.1.2. To determine which internal representations are most discriminative for Italian MGT detection, we evaluated ModernBERT-base using three distinct configurations (Table 4): the first layer, a specific block of intermediate layers, and the final output embedding. This investigation reveals that while detection performance improves with depth, distinct linguistic signals contribute differently to the model’s decision boundary.

Table 4

Ablation study on representation depth and model size using local context (3-sentence chunks). All values are reported in percentages (%). Performance consistently improves with layer depth, confirming that high-level semantic features (Final Layer) are more discriminative than surface or syntactic ones.

Model	Configuration	Acc	P	R	F1
ModernBERT-base	Layer 1 (Mean)	81.23	86.29	81.23	83.68
ModernBERT-base	Intermediate Layers 11-14	84.63	88.18	84.63	86.37
ModernBERT-base	Final Layer [CLS]	87.15	89.70	87.15	88.41

We selected **Layer 1** to test the hypothesis that MGT detection might be solvable via low-level surface artifacts alone. Using this layer yields an accuracy of 81.23%. While significantly above random chance, it lags behind deeper representations, suggesting that while surface signals are computationally cheap to extract, they are insufficient for high-accuracy detection.

The selection of **Layers 11-14** (84.63%) was derived from a specific linguistic hypothesis regarding “English-mediated” generation. Recent work on multilingual LLMs suggests that outputs are often syntactically “English-accented” (reflecting English) influenced grammatical preferences and distributional artifacts even when the surface language is Italian. Guo et al. [20] demonstrate that this accent manifests as systematic shifts in dependency relations and hierarchical substructures (e.g., tree configurations), while mechanistic studies suggest an English-centered latent space for semantic decisions [30]. If machine-generated text exhibits such structural rigidity, a detector should be able to recover these signals from the encoder’s intermediate representations.

To investigate the internal linguistic representations of ModernBERT-base, we adopt the layer-wise probing methodology of Conneau et al. [31] and evaluate seven diagnostic tasks spanning three linguistic dimensions⁵. For surface features, we use (1) *Length*, which predicts the total number of words to assess retention of simple information, and (2) *BigramShift*, which tests sensitivity to local word order by detecting whether adjacent tokens have been swapped. For syntactic features, we include tasks targeting both *vertical* and *horizontal* structure: (3) *Depth*, which estimates the maximum depth of the syntactic parse tree as a measure of hierarchical complexity, and (4) *TopConst*, which predicts the sequence of top-level constituents (e.g., NP–VP) to capture global syntactic organization. Finally, for semantic features, we consider (5) *Tense*, which identifies the grammatical tense of the main verb (e.g., past vs. present); (6) *SubjNum* and (7) *ObjNum*, which detect the grammatical number (singular/plural) of the subject and direct object, respectively.

Using the *Control Task* framework to ensure selectivity [32], our analysis revealed a clear functional stratification within ModernBERT, as shown in Table 5. While surface features like sentence length are accessible throughout the network and semantic tasks (*Tense*, *SubjNum*) peak in the final layers, it’s possible to identify a distinct “Syntactic Block” in the middle layers (11–14) where structural performance maximizes, similarly to organization of the original BERT model [33].

This middle block excels at *TopConst* (peaking at Layer 14 with selectivity 0.744), suggesting strong access to coarse phrase-structural configuration. This explains why extracting representations from Layers 11–14 outperforms Layer 1: the model successfully captures the linear organization of constituents, a key dimension where translation artifacts often appear. However, the probing analysis also highlights

⁵<https://github.com/facebookresearch/SentEval>

Table 5

Full Layer-wise Probing Results (Selectivity) for ModernBERT-base across all 22 layers. Values represent *Selectivity*, defined as the difference between the probe’s accuracy on the target linguistic task and a randomized control task, that is $S = Acc_{real} - Acc_{control}$. Note the distinct functional stratification: **Surface features** (i.e. the Text Length) impact positively ; **Syntactic features** (TopConst) show a peak in the middle block (Layers 11-14), justifying our ablation choice; while **Semantic features** (Tense, SubjNum) generally maximize in the deeper layers (18+).

Layer	Surface		Syntax		Semantics		
	<i>Length</i>	<i>BigramShift</i>	<i>Depth</i>	<i>TopConst</i>	<i>Tense</i>	<i>SubjNum</i>	<i>ObjNum</i>
0	0.767	-0.004	0.198	0.680	0.370	0.329	0.333
1	0.759	0.014	0.207	0.680	0.370	0.334	0.329
2	0.740	0.231	0.225	0.714	0.365	0.334	0.322
3	0.735	0.275	0.233	0.723	0.368	0.346	0.329
4	0.721	0.262	0.232	0.719	0.363	0.336	0.333
5	0.745	0.277	0.231	0.723	0.362	0.337	0.333
6	0.754	0.296	0.237	0.730	0.377	0.352	0.334
7	0.748	0.294	0.235	0.732	0.377	0.347	0.342
8	0.747	0.312	0.246	0.735	0.382	0.353	0.342
9	0.757	0.326	0.256	0.735	0.385	0.368	0.337
10	0.786	0.324	0.257	0.734	0.390	0.363	0.345
11	0.793	0.338	0.255	0.738	0.390	0.364	0.346
12	0.789	0.345	0.267	0.741	0.387	0.356	0.338
13	0.788	0.352	0.267	0.743	0.385	0.352	0.344
14	0.784	0.362	0.276	0.744	0.385	0.361	0.349
15	0.784	0.358	0.274	0.743	0.387	0.363	0.351
16	0.769	0.360	0.278	0.738	0.391	0.362	0.350
17	0.773	0.362	0.267	0.740	0.385	0.359	0.349
18	0.754	0.365	0.271	0.739	0.381	0.381	0.348
19	0.755	0.364	0.264	0.739	0.385	0.379	0.347
20	0.744	0.374	0.252	0.731	0.389	0.370	0.344
21	0.735	0.368	0.247	0.726	0.388	0.369	0.335
22	0.721	0.372	0.239	0.710	0.380	0.369	0.344

a critical limitation: the model struggles with syntactic *Depth* (Selectivity ≈ 0.27). Although selectivity for tree depth also peaks in this middle block, the absolute signal is weak. This suggests that while ModernBERT captures horizontal syntax, it does not preserve *hierarchical complexity* (vertical depth) in a form that remains linearly recoverable after mean pooling. Since human writing is often structurally “spikier” (greater variance in syntactic depth) than the smoother output of LLMs, the inability to fully leverage this vertical signal may explain why the syntactic block (84.63%) still underperforms compared to the final layer (87.15%), which aggregates syntactic and semantic information.

GSI:detect at EVALITA 2026: Overview of the Task on Detecting Gender Stereotypes in Italian^{*}

Gloria Comandini^{1,*}, Manuela Speranza², Sofia Brenna^{2,3}, Davide Testa^{2,4},
Stefania Cavagnoli⁵ and Bernardo Magnini²

¹*Italian Institute of Germanic Studies (IISG), Rome, Italy*

²*Fondazione Bruno Kessler (FBK), Trento, Italy*

³*Free University of Bozen-Bolzano, Bolzano, Italy*

⁴*University of Rome La Sapienza, Rome, Italy*

⁵*University of Rome Tor Vergata, Rome, Italy*

Abstract

GSI:detect is a new shared task for the recognition and classification of gender stereotypes (GSs) presented at EVALITA 2026. The task adopts a perspectivist approach in order to enhance the high subjectivity of GS recognition and analysis on a dataset of challenging short texts in Italian. GSI:detect is organized in: A) a Main Task (GS Detection) in which systems have to assign to a text the GS value, a numerical score that quantifies the extent to which a given text exhibits or refers to a GS; B) an optional Subtask (GS Classification) in which systems, given six pre-defined categories (e.g. role, relational, etc.) must assign one to each text. Seven teams from academic and non-academic environments took part in the challenge, with a total of 50 submitted runs for the Main Task and a total of 43 submitted runs for the optional Subtask. We present here first an overview of the GSI:detect task, the dataset and the evaluation criteria, then outline and discuss the participants' results. **Content warning:** Examples taken from the GSI:detect dataset may contain sensitive, offensive, or otherwise distressing content.

Keywords

gender stereotypes, perspectivism, linguistic resource, evaluation, LLMs,

1. Introduction and Motivation

The GSI:detect Task, organised within EVALITA 2026 [1], aims to take one step further in the state-of-the-art detection of gender stereotypes (GSs)¹. Gender stereotypes have recently been the object of extensive research in the context of automatic recognition of GSs in misogynistic hate speech [3, 4, 5] and also as far as large language models' production of stereotyped and biased material is concerned in text generation [6, 7] and translation [8].

However, with GSI:detect we aimed to first expand our focus beyond the context of hate speech, because GSs can also appear in non-hateful communication (even as a compliment: "women's nurturing nature will save the world!"). In fact, GSs are sometimes produced even by their own targets (e.g., a man who says "You know, men only have one thing in mind", or a woman who says "I failed in the math exam. Oh well, girls don't do well in math anyway, ah-ah"), who have internalized these biased views on gender.

Secondly, we want to underline that the recognition and therefore the analysis of stereotypes can be deeply subjective, as seen for example in the low inter-annotator agreement (IAA) (0.41, Cohen's

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

^{*}Corresponding author

✉ comandini@studigermanici.it (G. Comandini); manspera@fbk.eu (M. Speranza); sbrenna@fbk.eu (S. Brenna); dtesta@fbk.eu (D. Testa); stefania.cavagnoli@uniroma2.it (S. Cavagnoli); magnini@fbk.eu (B. Magnini)

🌐 <https://huggingface.co/GloriaComandini> (G. Comandini); <https://linktr.ee/davide.testa> (D. Testa)

🆔 0000-0003-3406-2819 (G. Comandini); 0009-0001-3748-1448 (S. Brenna); 0009-0002-2489-5323 (D. Testa);

0000-0003-1677-6455 (S. Cavagnoli); 0000-0002-0740-5778 (B. Magnini)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹We intend GSs as socially constructed beliefs about the 'appropriate' roles, behaviours, and appearances of a person regarding their gender; although GSI:detect focuses on GSs about men and women for the sake of simplicity, we describe gender as "a nonbinary construct" [2] and we intend to include GSs regarding non-binary people in future research.

k) in the recognition of racist stereotypes by [9]. This situation is common to most highly subjective NLP tasks, such as hate speech recognition [10], irony detection [9] or sentiment analysis applied to complex texts (e.g. newspaper articles or latin poetry) [11, 12, 13], where annotators judgments can be influenced by several factors, such as agreement or disagreement with a statement, the fact that the annotator belongs to the group targeted by hate or stereotypes [10], personal opinions [14] and interiorized biases [15, 16].

For all these reasons, we decided to adopt a Perspectivist Approach [17, 15, 18, 19, 16], in order to treat the inherent diversity in judgments and perceptions as a valuable asset, and not as a problem to solve. In fact, it has been observed that the classical methods of majority voting used to create gold-standard corpora may result in the suppression of perfectly valid points of view, which may even add interesting information about a task [14].

2. Task Description

GSI:detect aims to evaluate systems' ability to detect and classify GSs across different kinds of short texts; it includes both a compulsory main task and an optional sub-task.

2.1. Main Task: Gender Stereotype Detection

GS value	Example
0	Non comprendo come si possano paragonare due fenomeni, gravissimi entrambi e concordo, come femminicidi e morti sul lavoro. (<i>I don't understand how one can compare two phenomena, both very serious and I agree, such as feminicides and deaths at work.</i>)
0.25	Tenete duro ancora qualche giorno e i vostri fidanzati partiranno in vacanza con le loro mogli. (<i>Hold on for a few more days and your boyfriends will be going on vacation with their wives.</i>)
0.50	Io rimango dell'idea che un figlio ha sempre bisogno della sua mamma, anche per dire buongiorno e buona notte. E la mamma idem. Soprattutto la mamma (<i>I still think an [adult] child always needs his/her mother, even to say good morning and good night. And the mother too. Especially the mother</i>
0.75	[Commento ad articolo di giornale dal titolo "Negli Usa quasi un manager su due è donna. In Italia meno di 1 su 3"] Infatti il Made usa va' peggio del Made Italy (<i>[Comment on a newspaper article titled "In the US, almost one in two managers is a woman. In Italy, less than one in three"] In fact, Made in USA is doing worse than Made in Italy</i>)
1	[Rivolto a una utente donna] fatevi voi una doccia e copritevi. Le donne vere si coprono. Gli animali vanno in giro nudi. (<i>[Addressed to a female user] Take a shower and cover up. Real women cover up. Animals walk around naked.</i>)

Table 1

Sample texts annotated with GS values.

Given a short text, the main task requires systems to assign to it a numerical score, the GS value, that quantifies the degree to which the text contains or refers to a gender stereotype.

This is formulated as a regression task in which GS values are real numbers in the range [0,1], where 1 indicates the maximum degree of stereotypical content (the granularity is fixed at two decimal places). A few examples of short texts annotated with different GS values are reported in Table 1.

Notice that we consider two types of texts:

- **NO CONTEXT:** texts that can be understood without any additional contextual information (see the first three examples in Table 1);
- **WITH CONTEXT:** texts that are not self contained and are therefore enriched with contextual information in the form of standardized metadata (see the last two examples in Table 1).

2.2. Subtask: Gender Stereotype Classification

The subtask on GS classification is formulated as a multi-class classification task where, given a short text, systems are required to assign a GS category to that text; assignment must be performed for each

single text, independently of the GS value assigned by the systems in the main Task. Participation in the subtask was not compulsory; still, participants were strongly encouraged to submit their results to allow for a more comprehensive evaluation of the phenomenon.

Category	Example
ROLE	Cento uomini possono creare un accampamento, ma serve una donna per fare una casa. (<i>A hundred men can build a camp, but it takes a woman to make a home.</i>)
PERSONALITY	Sentivo qualcosa di speciale e sai, una donna non sbaglia mai le sensazioni. (<i>I felt something special and you know, a woman never mistakes her feelings.</i>)
COMPETENCE	[Commento ad articolo con titolo "La pilota della British Airways ubriaca in volo: cacciata dall'aereo, aggredisce pure i poliziotti"] Come si possono affidare le sorti di un aereo ad una donna?scherzo, naturalmente..." ([Comment on an article titled "British Airways pilot drunk on flight: kicked off plane, she even attacks police"] <i>How can you trust a plane's fate to a woman?just kidding, of course...</i>)
PHYSICAL	"Oppure c'hanno le 5^, vanno in giro scollate come i manifesti messi d'inverno,e poi se rimani ""attirato"" dalle loro protuberanze ci rimangono male Povere cucciole." ("Or they are a size D, they walk around with low-cut clothes like winter posters, and then if you get ""attracted"" by their protuberances, they get upset. Poor little things.")
SEXUAL	[Rivolto a una utente donna] fatevi voi una doccia e copritevi. Le donne vere si coprono. Gli animali vanno in giro nudi. ([Addressed to a female user] <i>Take a shower and cover up. Real women cover up. Animals walk around naked.</i>)
RELATIONAL	[Commento a meme con testo "Aspettavo che mi mandassi tu un messaggio" e sotto l'immagine di un uomo vestito da principessa] Tipico post da zitella ([Comment on a meme with the text "I was waiting for you to text me" and underneath a picture of a man dressed as a princess] <i>Typical spinster post</i>)

Table 2

Examples of texts assigned to the different GS categories.

The classification we propose, developed to capture the variety of ways in which stereotypes manifest in language and to support both linguistic analysis and automatic detection tasks, foresees the following GSs typologies (examples of texts assigned to the different categories are provided in Table 2)²:

- **ROLE** stereotypes: social and cultural expectations about what women and men should do and about how they should be;
- **PERSONALITY** stereotypes: emotional and behavioural traits assigned to men and women based on their gender;
- **COMPETENCE** stereotypes: generalized judgments of a person's abilities based on their gender;
- **PHYSICAL** stereotypes: expectations about the physical appearance of men and (especially) women, and all aspects of personal care in general;
- **SEXUAL** stereotypes: attitude and behaviour that men and women should have regarding sexuality;
- **RELATIONAL** stereotypes: the way in which women and men should behave in interpersonal/sentimental relations.

For a more accurate understanding of the task, participants were able to refer to the official guidelines for stereotype classification, which were followed during the manual annotation of the dataset.³

3. Dataset

The GSI:detect dataset⁴ consists of 1,010 short written texts in Italian (for a total of 52,118 tokens), collected from social media and informative websites.

The texts have been manually collected from a diverse array of online spaces to provide a balanced representation of formal and informal written Italian, a variety that also allows us to explore the theme of GSs in different contexts. In fact, they include excerpts from information websites, as well as users' comments from Facebook, Instagram and Reddit pages and groups discussing both gender-related issues

²Note that this taxonomy is not intended to be fully exhaustive, as it is derived from an abstraction over the direct observation of the examples contained in the dataset.

³The annotation guidelines are available for download at this link.

⁴The GSI:detect dataset is distributed under a CC BY-NC-SA 4.0 Licence. The dataset is publicly available at this link. The distributed dataset includes, besides the GS values, also the individual, non-aggregated labels assigned by all annotators, in order to enable systems to learn from annotator disagreement [19].

and more generic topics (e.g. feminist influencers, pick-up artists, “mom influencers”, parodic pages, math groups, gossip pages, major Italian newspapers, etc.),

Furthermore, the dataset includes GSs about not only women, but also men, or both, including texts that clearly express GSs as well as non-stereotypical examples, as well as cases where recognizing GSs may be a question of personal opinion and sensitivity.

3.1. Data Annotation

The GSI:detect dataset has been manually annotated by four expert annotators.

GS Value Annotation. Although all four annotators were expert and followed the annotation guidelines specifically created for GSI:detect (as mentioned in Section 2.2), the inherent subjectivity of the task inevitably introduced a certain level of disagreement. Following the perspectivist approach introduced in Section 1, we opted for merging all annotations into a numerical GS value, rather than selecting a binary label obtained through annotation aggregation on the basis of majority voting. This choice aligns with recent findings which indicate that leveraging disagreement is more convenient than effortlessly trying to eliminate it [15, 16].

The overall annotation procedure consists of two steps: (i) each annotator manually assigns, for each short text, a binary label *yes/no* indicating whether or not the text contains or refers to a GS; (ii) the final GS value is computed by combining the four individual annotations. The underlying assumption is that full IAA (all four annotators choose the label *no* or the label *yes*) corresponds to the endpoints of the continuum, while disagreement between annotators indicates intermediate GS values, such as 0.25 (three *no* labels and one *yes* label), 0.5 (two *yes* labels and two *no* labels), and 0.75 (three *yes* labels and one *no* label).

The overall IAA between the four annotators on the choice of the *yes* or *no* label is 0.61 (Fleiss’ k), which is a moderate agreement, common in highly subjective tasks [20], as seen in Section 1.

GS Category Annotation. When annotators identify a text as containing or referring to a GS, they additionally assign it to one of the six GS categories, following the classification outlined in the annotation guidelines and summarized in Section 2.2. Due to the more explorative nature of GS category annotation compared to the preceding annotation level, we adopted a more conventional strategy: in case of disagreement between the annotators, a single category is determined by majority vote, with ties resolved by a GS expert acting as a super-judge (required for 6% of the dataset). Regarding the GS category annotation, the four experts scored moderate agreement [20], with a IAA of 0.61 (Fleiss’ k).

3.2. Data Statistics: Test and Development Split

The complete dataset is divided as follows: 80% is allocated to the test set for the official evaluation and ranking of participant systems, while the remaining 20% constitutes the development data (dev set) (refer to Table 3 for more details). These proportions were chosen to balance the need for adequate data for model tuning with the goal of maintaining a larger and more representative test set for the final evaluation.

Table 3 reports also detailed information about the size of the dataset in terms of tokens⁵. In both the dev and test sets, approximately 58% of texts are WITH CONTEXT and 41% are NO CONTEXT (see Section 2.1). In the creation of the dev and test sets, particular care was taken to ensure a balanced distribution of examples across both subsets, as shown in Table 4a. Therefore, the

	Dev set	Test set	Total
WITH CONTEXT texts	82	323	405
NO CONTEXT texts	118	487	605
All Texts	200	810	1010
Tokens	10,055	42,063	52,118
Av. Length	50.27	51.93	51.6

Table 3: Dataset’s statistics.

⁵The token count was computed using the Italian rule-based tokenizer included in the *spaCy* library (<https://spacy.io>, version 3.8.7) as part of the *it_core_news_sm* linguistic model. The average length of texts is 51.6 tokens.

GS value	Dev set	Test set	Total	Total%
0	60	242	302	29.90%
0.25	25	84	109	10.79%
0.50	27	85	112	11.09%
0.75	25	105	130	12.87%
1	63	294	357	35.35%
	200	810	1010	

(a) Dataset distribution by GS value.

Category	Dev set	Test set	Total	Total%
Role	30	107	137	13.56%
Personality	29	108	137	13.56%
Competence	34	120	154	15.25%
Physical	20	90	110	10.89%
Sexual	14	72	86	8.52%
Relational	13	71	84	8.32%
GS value = 0	60	242	302	29.90%
	200	810	1010	

(b) Dataset distribution by GS category.

Table 4

Dataset distribution statistics.

split preserves the original distribution of GS

values, thereby guaranteeing a consistent representation of varying degrees of stereotypicality in both subsets. A comparable level of balance is also observed for GS categories (see Table 4b).

This careful selection ensures that both subsets are representative of the overall GSI:detect dataset, preventing unintended biases in the distribution of categories.

4. Evaluation

Evaluation in GSI:detect is designed to reflect the specific nature of both the main task on GS detection, formulated as a regression problem, and the subtask on GS classification, formulated as a multi-class classification problem. Accordingly, we adopt task-specific evaluation criteria to ensure meaningful comparison and reliable system ranking.

GS Detection. Participant systems’ performance is assessed using a score derived from the *Mean Squared Error* (MSE), which measures the average squared distance between predicted and annotated GS values, penalizing larger deviations more heavily. To improve interpretability and comparability across systems, the MSE is normalized with respect to the variance of the target distribution (*Normalized Mean Squared Error*, NMSE). Since lower NMSE values indicate better performance, we further transform this quantity into a bounded score defined as $\frac{1}{1+NMSE}$ so that higher values correspond to better predictive accuracy. This formulation enables an intuitive ranking of systems while preserving the relative performance differences.

In addition to the scores described above, we also report the *Concordance Correlation Coefficient* (CCC) as a complementary measure of agreement between predictions and reference values, capturing both correlation and potential systematic bias.

GS Classification. GS classification is evaluated using the *F1* score, which combines precision and recall into a single performance indicator.⁶ To account for possible class imbalance while maintaining sensitivity to per-class behaviour, we report both *Macro F1*, which weights all categories equally regardless of their frequency, and *Micro F1*, which reflects the overall performance at instance-. While both measures are reported for analysis purposes, *Micro F1* is adopted as the official metric for system ranking, as it provides an instance-level estimate of overall classification performance.

Baselines. For both GS detection and classification, participant systems’ performance is compared against a set of four baselines⁷. They include a simple heuristic baseline obtained by assigning a constant

⁶As annotators assigned a GS category only to texts containing or referring to a stereotype, texts with GS value = 0 (i.e. no-no-no-no annotation) don’t have a category. In the GS classification subtask, systems’ performance has been therefore evaluated on 568 out of 810 texts.

⁷Baselines were computed only for the zero-shot track.

Team Name	GS Detection				GS Classification			
	Zero-shot	Few-shot	Fine-Tuning	Encoder-only	Zero-shot	Few-shot	Fine-Tuning	Encoder-only
DIAG-Sapienza [21]	1	1	-	4	1	1	-	4
Festa [22]	5	5	-	5	5	5	-	5
MINDS [23]	-	2	-	-	-	-	-	-
Prisma [24]	5	1	-	-	5	1	-	-
StereoBusters [24]	5	5	-	-	5	5	-	-
Tiz [25]	5	-	5	-	5	-	-	-
VellaAsta ⁸	-	-	-	1	-	-	-	1
Total	21	14	5	10	21	12	0	10

Table 5

Number of runs submitted by each team for each track in the GS Detection and GS Classification tasks.

GS value of 0.5 and a random GS category prediction (B1 in Table 6), as well as the performance of two LLMs, namely *Qwen3-14B* (B2), and *GPT-5-nano-2025-08-07*. The performance of *GPT-5*, in particular, is evaluated under two prompting configurations: one where GS detection and classification are jointly addressed within a single prompt (B3), and one where the tasks are solved independently with two separate prompts (B4).

5. Participants

The GSI:detect shared-task attracted the participation of seven teams, coming both from academic and non-academic environments. Participant were allowed to submit multiple runs for each task, exploring different model architectures, prompting strategies, and technical configurations. The evaluation campaign included four different tracks for both the main task and the subtask: *zero-shot*, *few-shot*, *fine-tuning* of LLMs, and *encoder-only models*. Not all teams submitted runs to all tracks and tasks. Table 5 presents the participating teams and reports the number of runs submitted by each of them for each track in both GS Detection and GS Classification tasks. Further analysis of the impact of the different runs and of their behavior across the two tasks, leading to different performance trends, is presented in Section 6. For a detailed description of the individual system configurations and methodologies associated with each run, we refer the reader to the participants’ reports, cited in Table 5.

6. Results and Discussion

The results are organized according to the four main tracks of the shared task, under which participants submitted multiple runs for both the main task and the subtask. This experimental setting results in eight distinct rankings, corresponding to the combination of the two tasks and the four tracks.

6.1. Zero-Shot Track

Tables 6 and 7 report the scores obtained by the participating systems on the main task and on the subtask, respectively, in the zero-shot setting. Overall, in both tasks several participant systems outperform the proposed baselines, which are positioned approximately in the middle of the ranking and therefore act as a rough boundary between higher- and lower-performing approaches in this track.

For our main task, the best-performing system is submitted by the DIAG-Sapienza team [21], based on *GPT-5* in a configuration that generates together the predicted GS value, the GS category plus a one-sentence explanation within the same model’s call (see run 1 in Table 6 and 7). Interestingly, while this model’s configuration achieves the top rank in GS Detection, its performance drops by seven positions in GS Classification, highlighting the increased difficulty of fine-grained stereotype categorization and

⁸As the VellaAsta team correctly and timely submitted the output of their system for official evaluation, we present their results even if they did not submit a report describing the system.

of the underlying reasoning process. Nevertheless, in the main task this configuration substantially outperforms our *GPT-5 nano* baseline (i.e., B4), whereas in the classification task the performance gap becomes much smaller (0.58 Micro F1 for DIAG-Sapienza vs. 0.53 for B4).

Another team showing consistently strong performance in this track is StereoBusters [26], which evaluates several models of different sizes and configurations. In particular, *Llama 3.3* (70B) achieves competitive results in GS Detection, approaching the performance of the closed-source *GPT-5* model by DIAG-Sapienza (0.63 vs. 0.70). Moreover, *Llama 3.3* consistently maintains a high level of performance also in GS Classification (0.64 Micro F1), outperforming both proprietary models (*GPT*- and *Gemini*-based) and the systems submitted by the other teams. Additionally, the ensemble strategy adopted by this team (i.e., run 4), combining the predictions of four LLMs to determine the stereotype category, proves particularly effective in the classification task, achieving the highest Micro F1 score (0.646) in this subtask, with an improvement of roughly ten positions in the ranking compared to the main task.

The Festa team [22] explores multiple configurations of *Gemini 2.5 Flash*, outperforming most of our baselines in both tasks. For GS Detection, the best-performing configuration (i.e., run 4) relies on an English prompt, despite the Italian nature of the data, combined with negative constraints aimed at minimizing false positives. By contrast, the use of Chain-of-Thought prompting both in English and Italian (runs 1 and 3) appears less beneficial in this task, yielding scores approximately seven points lower than run 4 (0.62 vs. 0.55). In the classification task, however, Chain-of-Thought prompting proves more effective, with performance comparable to *Llama 3.3* (70B) by StereoBusters. This suggests that explicitly encouraging intermediate reasoning steps may support finer-grained category discrimination.

Finally, in this setting, the systems submitted by the Tiz [25] and Prisma [24] teams show substantially lower performance, consistently falling below the LLM-based baselines (i.e., B2, B3, and B4), with this performance gap remaining consistent across both the main task and the subtask. In particular, Prisma explores multiple configurations of *Claude 3.5 Sonnet* based on different annotator personas and their aggregation; however, these highly polarized configurations do not yield competitive results, possibly due to a mismatch between the induced persona biases in the configuration and the annotator perspectives underlying the dataset. This could also suggest that strong persona conditioning, when misaligned with the annotator distributions of the dataset, may introduce biases that increase the distance from the target judgments.

Overall, what emerges from this scenario is that, although *GPT-5* (DIAG-Sapienza) represents the best-performing system in the main GS Detection task, open-source models with different configurations and modeling strategies are able to closely approach, match, or even surpass closed-source systems across both tasks. In particular, StereoBusters’ open-source models narrow the gap with *GPT-5* in the main task and outperform both proprietary systems and our baselines in GS Classification, regardless of whether large-scale (*Llama 3.3* 70B) or mid-sized (*Gemma 3* 12B) models are used. This trend suggests – and confirms – that stereotype categorization, which inherently involves subjective interpretation and nuanced reasoning, benefits less from pure model scale and more from diversified modeling choices and decision aggregation strategies.

6.2. Few-shot Track

Most of the teams participating in the zero-shot track also submitted systems to the few-shot track, except for Tiz team, which participated exclusively in the former setting. This semi-overlap allows for a direct comparison of model behaviour across the two settings, highlighting how the same architectures can exhibit substantially different performance when provided with in-context examples.

According to Table 8, for the GS Detection task, the DIAG-Sapienza team again achieves the best performance with *GPT-5* (i.e., run 1, four-shot), followed by the family of *Gemma 3* models (12B and 27B) evaluated by StereoBusters. However, the systems developed by both teams are not able to maintain the same trend in the subtask (see Table 9), dropping several positions in the ranking, except for *Gemma 3* (27B). This finding suggests that, within the same model family, model size may positively influence performance across both tasks.

Notably, in contrast with the zero-shot setting, the Prisma team shows a marked improvement: its

Team name	Model	Run id	$1/(1 + NMSE) \uparrow$	MSE $\downarrow$	NMSE $\downarrow$	CCC $\uparrow$
DIAG-Sapienza	GPT-5	1	0.70	0.077	0.43	0.78
StereoBusters	Llama_3.3_70B	1	0.63	0.11	0.60	0.60
StereoBusters	Llama_3.3_70B	2	0.62	0.11	0.61	0.59
Festa	Gemini_2.5_Flash	4	0.62	0.11	0.61	0.70
Festa	Gemini_2.5_Flash	5	0.62	0.11	0.62	0.69
StereoBusters	Gemma_3_12B	0	0.61	0.11	0.64	0.56
Festa	Gemini_2.5_Flash	2	0.61	0.11	0.64	0.69
BASELINE	GPT-5 nano	B4	0.61	0.11	0.64	0.60
Tiz	Gemma_3_12B	5	0.59	0.12	0.69	0.63
BASELINE	GPT-5 nano	B3	0.59	0.12	0.69	0.57
StereoBusters	Panel_4LLMS	4	0.57	0.13	0.75	0.62
StereoBusters	Panel_4LLMs	3	0.56	0.14	0.77	0.60
Festa	Gemini_2.5_Flash	1	0.56	0.14	0.79	0.63
Festa	Gemini_2.5_Flash	3	0.55	0.14	0.80	0.62
BASELINE	Qwen-3_14B	B2	0.54	0.15	0.84	0.46
BASELINE	N/A	B1	0.50	0.18	1.01	0
Tiz	Gemma_3_12B	3	0.48	0.19	1.06	0.55
Tiz	Gemma_3_12B	2	0.48	0.19	1.07	0.54
Tiz	Gemma_3_12B	1	0.47	0.20	1.12	0.52
Tiz	Gemma_3_12B	4	0.43	0.24	1.33	0.42
Prisma	Claude_3.5_Sonnet	2	0.33	0.36	2.04	0.15
Prisma	Claude_3.5_Sonnet	4	0.32	0.38	2.11	0.08
Prisma	Claude_3.5_Sonnet	1	0.32	0.38	2.11	0.09
Prisma	Claude_3.5_Sonnet	5	0.31	0.40	2.23	0.09
Prisma	Claude_3.5_Sonnet	3	0.31	0.40	2.24	0.08

Table 6
Results for *zero-shot* track [Main Task].

Team name	Model	Run id	F1 Micro $\uparrow$	F1 Macro $\uparrow$
StereoBusters	Panel_4LLMS	4	0.65	0.64
StereoBusters	Llama_3.3_70B	1	0.64	0.64
StereoBusters	Panel_4LLMs	3	0.64	0.63
StereoBusters	Llama_3.3_70B	2	0.63	0.63
Festa	Gemini_2.5_Flash	1	0.60	0.56
Festa	Gemini_2.5_Flash	5	0.60	0.55
Festa	Gemini_2.5_Flash	4	0.59	0.55
Festa	Gemini_2.5_Flash	2	0.59	0.54
DIAG-Sapienza	GPT-5	1	0.58	0.52
Festa	Gemini_2.5_Flash	3	0.57	0.52
StereoBusters	Gemma_3_12B	0	0.54	0.53
BASELINE	GPT-5 nano	B4	0.53	0.52
BASELINE	GPT-5 nano	B3	0.52	0.50
BASELINE	Qwen-3_14B	B2	0.39	0.39
Tiz	Gemma_3_12B	2	0.23	0.23
Tiz	Gemma_3_12B	3	0.23	0.23
Tiz	Gemma_3_12B	1	0.22	0.22
Prisma	Claude_3.5_Sonnet	1	0.22	0.29
Tiz	Gemma_3_12B	5	0.22	0.22
Tiz	Gemma_3_12B	4	0.21	0.21
BASELINE	N/A	B1	0.18	0.18
Prisma	Claude_3.5_Sonnet	4	0.18	0.23
Prisma	Claude_3.5_Sonnet	2	0.17	0.24
Prisma	Claude_3.5_Sonnet	3	0.13	0.18
Prisma	Claude_3.5_Sonnet	5	0.12	0.17

Table 7
Results for *zero-shot* track [Subtask].

system (i.e., run 6) benefits significantly from this configuration based on selecting 5 examples per category through stratified sampling. In this case, *Claude 3.5 Sonnet* appears to react positively both to the presence of in-context examples and to the inclusion of additional information derived from the annotation guidelines. This effect is even more evident in the GS Classification task (Table 9), where the same configuration outperforms all other submissions by approximately ten points (0.71).

Yet, *Gemini 2.5 Flash* exhibits a more heterogeneous behavior: while a 14-shot configuration leads to one of the worst performance in GS Detection, the presence of examples proves beneficial for reasoning on GS Classification, but the improvement over the zero-shot setting remains marginal (0.67 vs. 0.60).

A different trend is observed for the MINDS team [23], which evaluates *Qwen 2.5* (14B) exclusively in the few-shot setting for GS Detection. Two configurations (runs 1 and 2), each using 20 in-context

Team name	Model	Run id	Shots	$1/(1 + NMSE) \uparrow$	MSE $\downarrow$	NMSE $\downarrow$	CCC $\uparrow$
DIAG-Sapienza	GPT-5	1	4	0.68	0.08	0.46	0.76
StereoBusters	Gemma_3_27B	2	5	0.64	0.10	0.55	0.65
StereoBusters	Gemma_3_12B	0	5	0.63	0.10	0.59	0.60
StereoBusters	Gemma_3_12B	1	5	0.61	0.11	0.63	0.61
Prisma	Claude_3.5_Sonnet	6	35	0.59	0.12	0.70	0.63
StereoBusters	Panel_4LLMs	4	5	0.59	0.13	0.71	0.65
MINDS	Qwen2.5_14B	1	20	0.58	0.13	0.72	0.46
Festa	Gemini_2.5 Flash	5	non-fixed	0.57	0.13	0.75	0.63
Festa	Gemini_2.5 Flash	1	12	0.56	0.14	0.77	0.62
MINDS	Qwen2.5_14B	2	20	0.56	0.14	0.77	0.42
Festa	Gemini_2.5 Flash	2	10	0.56	0.14	0.78	0.62
Festa	Gemini_2.5 Flash	4	14	0.55	0.14	0.82	0.60
StereoBusters	Panel_4LLMs	3	5	0.55	0.15	0.82	0.56
Festa	Gemini_2.5 Flash	3	6	0.50	0.18	1.00	0.55

Table 8
Results for *few-shot* track [Main Task].

Team name	Model	Run id	Shot	F1 Micro $\uparrow$	F1 Macro $\uparrow$
Prisma	Claude_3.5_Sonnet	6	35	0.71	0.61
Festa	Gemini_2.5 Flash	4	14	0.67	0.67
StereoBusters	Gemma_3_27B	2	5	0.67	0.66
Festa	Gemini_2.5 Flash	5	non-fixed	0.67	0.66
StereoBusters	Panel_4LLMs	3	5	0.66	0.65
StereoBusters	Panel_4LLMs	4	5	0.66	0.65
Festa	Gemini_2.5 Flash	1	12	0.66	0.65
Festa	Gemini_2.5 Flash	2	10	0.65	0.63
Festa	Gemini_2.5 Flash	3	6	0.64	0.63
DIAG-Sapienza	GPT-5	1	4	0.61	0.55
StereoBusters	Gemma_3_12B	0	5	0.57	0.57
StereoBusters	Gemma_3_12B	1	5	0.56	0.56

Table 9
Results for *few-shot* track [Subtask].

examples, extract logits from the model and feed them into downstream statistical predictors (i.e., Linear Regression and KNN) to estimate the numerical GS value. This hybrid approach, however, remains substantially poor, with a gap of nearly ten points compared to the top model in this track (0.58 vs. 0.68).

Moreover, the trend shown here closely mirrors the zero-shot setting one, further confirming that performance gains are driven less by model scale alone and more by how in-context examples are selected, structured, and integrated into the reasoning process. In support of this observation, we can observe that i. *GPT-5* achieves the highest performance in GS Detection, yet its performance drops sharply in GS Classification (ten positions), despite the use of a four-shot prompt, ii. the perspectivist approach adopted by Prisma in the GS Classification subtask, combined with a careful selection of representative examples for each category, proves particularly effective, iii. in the main task, an open-source mid-sized model such as *Gemma 3* (27B) by StereoBusters is able to achieve performance comparable to that of *GPT-5* (DIAG-Sapienza).

6.3. Fine-tuning

Although potentially interesting for comparison with the other tracks, this setting was explored only by the Tiz team, exclusively through different configurations of the same model (i.e., *Gemma 3* 12B), and only for the main task. Nevertheless, the results obtained in this setup are highly promising, as the best-performing system (i.e., run 2) achieves a score of 0.64, ranking behind the best zero-shot and few-shot systems by only 6 and 2 points, respectively. Moreover, it is worth emphasizing that these fine-tuning results were obtained using an open model, which here is nearly matching the performance of a state-of-the-art proprietary model (i.e., *GPT-5*) under both of the aforementioned settings.

Team name	Model	Run id	$1/(1 + NMSE) \uparrow$	MSE $\downarrow$	NMSE $\downarrow$	CCC $\uparrow$
Tiz	Gemma_3_12B	2	0.64	0.10	0.57	0.63
Tiz	Gemma_3_12B	4	0.62	0.11	0.60	0.58
Tiz	Gemma_3_12B	1	0.62	0.11	0.61	0.61
Tiz	Gemma_3_12B	5	0.62	0.11	0.61	0.57
Tiz	Gemma_3_12B	3	0.60	0.12	0.67	0.52

Table 10
Results for *fine-tuning (LLMs)* track [Main Task].

Team name	Model	Run id	$1/(1 + NMSE) \uparrow$	MSE $\downarrow$	NMSE $\downarrow$	CCC $\uparrow$
Festa	UmBERTo	1	0.56	0.14	0.78	0.31
Festa	UmBERTo	2	0.56	0.14	0.78	0.31
Festa	UmBERTo	5	0.56	0.14	0.78	0.31
Festa	UmBERTo	4	0.55	0.14	0.81	0.36
Festa	UmBERTo	3	0.54	0.15	0.85	0.32
VellaAsta	tum-nlp/bertweet-sexism	1	0.49	0.19	1.05	0.24
DIAG-Sapienza	RoBERTa	4	0.47	0.20	1.13	0.37
DIAG-Sapienza	RoBERTa	3	0.46	0.21	1.16	0.37
DIAG-Sapienza	RoBERTa	1	0.45	0.21	1.21	0.36
DIAG-Sapienza	RoBERTa	2	0.45	0.22	1.24	0.36

Table 11
Results for *encoder-only models* track [Main Task].

6.4. Encoder-only models

The use of encoder-only models represents a valuable reference for analyzing discrimination-oriented tasks. Although such systems can often achieve performance comparable to or even better than LLMs, especially when fine-tuned on the target task [27], this trend does not emerge in our track as proved by the Festa, DIAG-Sapienza, and VellaAsta teams, reported in Tables 11 and 12. Overall, their results fall below our baselines on the main task and consistently find themselves in the lowest positions in the GS classification subtask. When comparing models within the same track, the BERT-based *UmBERTo* (Festa) achieves the best performance, with a clear margin over *RoBERTa* model (DIAG-Sapienza) and *bertweet-sexism*⁹ model (VellaAsta) likely due to its Italian-specific tokenization and masking strategies. Conversely, the *tum-nlp/bertweet-sexism* model outperforms *RoBERTa* in the GS detection task, as it has been fine-tuned for sexism detection on Twitter data. However, this specialization does not appear sufficient to achieve competitive performance on the more fine-grained GS classification task.

Overall, the consistently lower performance of encoder-only models across both tasks suggests that the limitations observed are not merely due to model capacity, but rather to architectural constraints. Unlike LLM-based systems, encoder-only models lack an explicit generative and reasoning component, which appears crucial for capturing the contextual, interpretative, and often implicit nature of gender stereotypes. While task-specific pre-training or fine-tuning (e.g., sexism detection on social media) can provide advantages in coarse-grained detection, as observed for *bertweet-sexism* model (VellaAsta Team) in the main task, such specialization does not translate into robust performance on fine-grained GS classification. This highlights the difficulty for encoder-only architectures to model subjective and socially grounded distinctions without access to richer contextual reasoning mechanisms.

7. Conclusions

In this paper, we presented the EVALITA shared task GSI:detect, which focuses on the automatic identification and classification of gender stereotypes in Italian. The task was structured into two subtasks: (i) a main task targeting the detection of GSs, and (ii) a fine-grained subtask aimed at classifying the type of GS expressed in the text. The dataset was constructed by explicitly leveraging disagreement among human annotators, to preserve the intrinsic subjectivity of the phenomenon rather than enforcing a single, fully convergent label. This design choice allows the benchmark to better

⁹This model can be found in the Huggingface Hub [here](#).

Team name	Model	Run id	F1 Micro ↑	F1 Macro ↑
Festa	UmBERTo	1	0.52	0.50
Festa	UmBERTo	2	0.49	0.47
Festa	UmBERTo	3	0.48	0.46
Festa	UmBERTo	5	0.47	0.45
Festa	UmBERTo	4	0.45	0.43
DIAG-Sapienza	RoBERTa	4	0.37	0.29
DIAG-Sapienza	RoBERTa	3	0.36	0.29
DIAG-Sapienza	RoBERTa	2	0.35	0.29
DIAG-Sapienza	RoBERTa	1	0.35	0.29
VellaAsta	tum-nlp/bertweet-sexism	1	0.10	0.11

Table 12

Results for *encoder-only models* track [Subtask].

reflect the variability of human judgments in sensitive and socially grounded tasks.

The results obtained by the seven participating teams on the main task, and by a subset of them on the subtask, highlight how different model architectures and configurations have diverse responses to a task in which subjectivity is a core component. Across zero-shot, few-shot, and encoder-only settings, we observe that performance is not solely determined by model scale, but is strongly influenced by architectural choices, prompting strategies, and the way contextual information is integrated into the reasoning process. In particular, for such fine-grained and inherently subjective settings, the performance gap between open-source and state-of-the-art proprietary systems consistently narrows, and in several cases reverses, especially when open models are combined with diversified perspectives or carefully selected in-context examples. Conversely, encoder-only architectures struggle to achieve competitive performance, suggesting that generative and reasoning capabilities play a crucial role in modeling socially grounded and interpretative distinctions.

These findings suggest that model behavior cannot be fully characterized only in terms of raw accuracy, but should also be analysed in relation to how models implicitly encode and reproduce subjective judgments. More broadly, the results highlight the importance of evaluation frameworks that explicitly account for subjectivity, disagreement, and modeling diversity when assessing AI systems on sensitive social phenomena. For this reason, investigating the interplay between model and human subjectivity represents a promising research direction for explaining some of the observed dynamics. Future work will focus on socio-profiling the models to relate their prediction patterns to specific socio-demographic groups, shedding light on the sources and structure of their biases and sensitivities.

Acknowledgments

This paper and the GSI:detect Task are the result of collaboration between all authors. Gloria Comandini wrote 1 and 3; Manuela Speranza wrote 2; Sofia Brenna wrote 4; Davide Testa wrote 5, 6 and 7.

This work was carried out while Davide Testa was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome together with Fondazione Bruno Kessler (FBK).

Declaration on Generative AI

Authors used ChatGPT (GPT-5.2) to refine the manuscript’s writing style and to support parts of the evaluation code. The authors reviewed and edited all outputs and take full responsibility for the content.

References

- [1] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [2] A. P. Association, Guidelines for psychological practice with transgender and gender nonconforming people, *American Psychologist* 70 (2015) 832–864.
- [3] E. Fersini, D. Nozza, P. Rosso, Overview of the evalita 2018 task on automatic misogyny identification (ami), in: T. Caselli, N. Novielli, V. Patti, P. Rosso (Eds.), *EVALITA Evaluation of NLP and Speech Tools for Italian. Proceedings of the Final Workshop*, Accademia University Press, Torino, 2018.
- [4] H. Kirk, W. Yin, B. Vidgen, P. Röttger, Semeval-2023 task 10: Explainable detection of online sexism, in: A. Ojha, A. Doğruöz, G. Da San Martino, H. Tayyar Madabushi, R. Kumar, E. Sartori (Eds.), *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Association for Computational Linguistics, Stroudsburg, 2023, p. 2193–2210.
- [5] L. Plaza, J. Carrillo-de Albornoz, R. Morante, E. Amigó, J. Gonzalo, D. Spina, P. Rosso, Overview of exist 2023: sexism identification in social networks, in: J. Kamps, L. Goeuriot, F. Crestani, M. Maistro, H. Joho, B. Davis, C. Gurrin, U. Kruschwitz, A. Caputo (Eds.), *Advances in Information Retrieval: 45th European Conference on Information Retrieval, ECIR 2023, Dublin, Ireland, April 2–6, 2023, Proceedings, Part III*, Springer, Cham, 2023, pp. 593–599.
- [6] Y. T. Cao, A. Sotnikova, H. Daumé III, R. Rudinger, L. Zou, Theorygrounded measurement of u.s. social stereotypes in english language models, in: *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, Seattle, 2022, p. 1276–1295.
- [7] A. Ovalle, P. Goyal, J. Dhamala, Z. Jagers, K.-W. Chang, A. Galstyan, R. Zemel, R. Gupta, “i’m fully who i am”: Towards centering transgender and non-binary voices to measure biases in open language generation., in: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, Association for Computing Machinery, New York, 2023, p. 1246–1266.
- [8] B. Savoldi, J. Bastings, L. Bentivogli, E. Vanmassenhove, A decade of gender bias in machine translation, *Patterns* 6 (2025) 101257.
- [9] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, S. M., An italian twitter corpus of hate speech against immigrants, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Paris, 2018, pp. 2798–2805.
- [10] M. Wojatzki, T. Horsmann, D. Gold, T. Zesch, Do women perceive hate differently: Examining the relationship between hate speech, gender, and agreement judgement, in: A. Barabresi, H. Biber, F. Neubarth, R. Osswald (Eds.), *Proceedings of the 14th conference on Natural Language Processing (KONVENS 2018)*, 2018, pp. 110–120.
- [11] L. Krusic, Constructing a sentiment-annotated corpus of austrian historical newspapers: Challenges, tools, and annotator experience, in: M. Härmäläinen, E. Öhman, S. Miyagawa, K. Alnajjar, Y. Bizzoni (Eds.), *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, Association for Computational Linguistics, 2024, pp. 51–62.
- [12] G. Comandini, «frena la locomotiva d’europa»: il ‘linguaggio della ferrovia’ per raccontare la crisi economica tedesca. analisi linguistica sul corpus locomin, *Studi Germanici* 27 (2025) 127–155.
- [13] R. Sprugnoli, F. Mambrini, M. Passarotti, G. Moretti, The sentiment of latin poetry. annotation and automatic analysis of the odes of horace, *Italian Journal of Computational Linguistics* 9 (2023).
- [14] M. Klenner, A. Göhrring, M. Amsler, Harmonization sometimes harms, in: S. Ebling, D. Tuggener, M. Hürlimann, M. Cieliebak, M. Volk (Eds.), *Proceedings of the 5th Swiss Text Analytics Conference (SwissText) 16th Conference on Natural Language Processing (KONVENS)*, CEUR Workshop Proceedings, 2020.
- [15] V. Basile, F. Cabitza, A. Campagner, Toward a perspectivist turn in ground truthing for predictive computing., in: *Toward a Perspectivist Turn in Ground Truthing for Predictive Computing*, Association for the Advancement of Artificial Intelligence, Washington DC, 2023, pp. 6860–6868.
- [16] B. Muscato, C. Sree Mala, M. Marchiori Manerba, G. Gezici, F. Giannotti, An overview of recent approaches to enable diversity in large language models through aligning with human perspectives, in: G. Abercrombie, V. Basile, D. Bernadi, S. Dudy, S. Frenda, L. Havens, S. Tonelli (Eds.), *Proceedings*

- of the 3rd Workshop on Perspectivist Approaches to NLP (NLPerspectives) @ LREC-COLING 2024, ELRA and ICCL, 2024, p. 49–55.
- [17] V. Basile, It’s the end of the gold standard as we know it. on the impact of pre-aggregation on the evaluation of highly subjective tasks, in: G. Vizzari, M. Palmonari, A. Orlandini (Eds.), Proceedings of the AIXIA 2020 Discussion Papers Workshop, CEUR Workshop Proceedings, 2020, pp. 31–40.
 - [18] G. Rizos, B. Schuller, Average jane, where art thou? – recent avenues in efficient machine learning under subjectivity uncertainty, in: M. Lesot, S. Vieira, M. Reformat, J. Carvalho, A. Wilbik, B. Bouchon-Meunier, R. Yager (Eds.), Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, Cham, 2020, pp. 42–55.
 - [19] M. Madeddu, S. Frenda, M. Lai, V. Patti, V. Basile, Disaggregating it corpus: A disaggregated Italian dataset of hate speech, in: F. Boschetti, G. E. Lebani, B. Magnini, N. Novielli (Eds.), Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023), CEUR Workshop Proceedings, 2023, pp. 243–250.
 - [20] R. Artstein, Inter-annotator agreement, in: N. Ide, J. Pustejovsky (Eds.), Handbook of Linguistic Annotation, Springer, Dordrecht, 2017, pp. 297–313.
 - [21] O. E. Sorokoletova, E. Musumeci, D. Nardi, Diag-sapienza at gsi:detect: A technical report, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [22] D. Festa, Festa at gsi:detect: In-context learning vs. fine-tuning for gender stereotype detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [23] F. Giobergia, Minds at gsi:detect: From logits to degrees of agreement in gender stereotype detection with llms, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [24] C. Zaghi, Prisma at gsi:detect: Comparing pbj persona-based and few-shot approaches to gender stereotype detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [25] T. Labruna, Tiz at gsi:detect: Modeling gender stereotypes detection as multi-category gender stereotype scoring, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [26] S. Greco, M. La Quatra, M. Marchiori Manerba, R. Muñoz Sánchez, A. T. Cignarella, Stereobusters at gsi:detect: Llm-based detection and human qualitative analysis of gender stereotypes in Italian short texts, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [27] P. Göttfert, R. Huber, F. Mariani, NLPeace@GermEval shared task 2025: Fine-tuned BERT vs. prompted LLMs for German hate speech detection, in: Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops, HsH Applied Academics, Hannover, Germany, 2025.

DIAG-Sapienza at GSI:detect: Joint Detection and Classification of Gender Stereotypes with Structured Prompting and Fine-Tuning

Olga E. Sorokoletova¹, Emanuele Musumeci¹ and Daniele Nardi¹

¹Sapienza University of Rome, Piazzale Aldo Moro, 5, Roma, 00185, Italy

Abstract

This report presents the contribution from the DIAG-Sapienza team for the GSI:detect task at EVALITA 2026. We address both the main regression task and the classification sub-task in a joint formulation, motivated by the hypothesis that category-level information can support more accurate estimation of gender stereotype intensity, especially in low-data settings. Our contribution features three systems corresponding to the challenge tracks: 1) a zero-shot approach that used GPT-5 as an LLM-as-a-Judge with structured prompting and explanation generation to elicit implicit reasoning, 2) a few-shot approach that augments prompting with randomly sampled and retrieved in-context examples, and 3) an encoder-only fine-tuned RoBERTa-based model that integrates LLM-generated reasoning sentences as additional input. Leaderboard results on the official test set show that our zero-shot and few-shot systems rank first and second, on the main regression task, outperforming all competing submissions. Overall, the results suggest that jointly predicting gender stereotype values and categories benefits the regression task, and that eliciting concise explanations can improve prediction quality.

Keywords

Gender Stereotype Detection, Bias Detection, Large Language Models, LLM-as-a-Judge, Zero-Shot Learning, Few-Shot Learning

1. Introduction

In this report, we address the GSI:detect challenge [1] of the EVALITA 2026 campaign [2], which focuses on evaluating systems' ability to detect and classify gender stereotypes (GSs) across diverse types of short texts in Italian. The dataset includes manually collected texts drawn from social media and informative websites, spanning both formal and informal language, which makes the task representative of real-world usage. The challenge comprises a compulsory main task on *Gender Stereotype Detection* and an optional sub-task on *Gender Stereotype Classification*. The main task is formulated as a regression problem, requiring the assignment of a numerical score in the range [0, 1] that quantifies the degree to which a text contains or refers to a gender stereotype. The sub-task is a multi-class classification problem, in which the system must assign one of six predefined Gender Stereotype categories: Role, Personality, Competence, Physical, Sexual, and Relational, when a stereotype is present, or None when no stereotype is detected.

In our approach, we address both tasks jointly rather than treating them as independent challenges. While a modular formulation is possible, we generate both outputs within the same system call from the same input. This design choice is motivated by observations in the literature that Large Language Models (LLMs) often struggle with purely numerical outputs and tend to perform less reliably on regression tasks than on text-based tasks [3]. We therefore hypothesize that explicitly incorporating information from the classification task into the context provided to the LLM, including the definitions of stereotype categories, can support the estimation of gender stereotype values and improve regression performance in zero-shot and few-shot settings.

This issue does not arise in fine-tuning scenarios, where the model is explicitly adapted to the target

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
✉ sorokoletova@diag.uniroma1.it (O. E. Sorokoletova); musumeci@diag.uniroma1.it (E. Musumeci); nardi@diag.uniroma1.it (D. Nardi)

🆔 0009-0005-4356-2649 (O. E. Sorokoletova); 0009-0004-2359-5032 (E. Musumeci); 0000-0001-6606-200X (D. Nardi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

objectives rather than being prompted to perform them as a by-product of next-word prediction, as in general-purpose LLMs. However, fine-tuned models are highly sensitive to architecture choices. In this setting, it is straightforward to implement a parallelized architectures that jointly handle regression and classification while sharing intermediate representations, for example, a shared encoder with task-specific heads. This design enables a controlled assessment of whether joint modeling yields measurable gains, and it allows us to test the extent to which shared representations can transfer information between the two tasks under supervised training.

These modeling choices must ultimately be grounded in a clear operationalization of what constitutes a stereotype in the first place. Within the GSI:detect challenge, a stereotype is defined as a pre-constituted, generalized, and simplistic opinion that is not based on the evaluation of individual cases but is mechanically repeated about people, events, or situations [4]. Gender stereotypes, in particular, are frequently found in misogynistic hate speech but also appear in non-hateful communication, where they may be unconscious and used with positive meaning. The greatest challenge lies in detecting such stereotypes when they are latent in both the form and the implied content of the analyzed text.

For this reason, in our zero-shot and few-shot experiments, we explicitly feature this definition in the task explanation provided to the LLM. This step is crucial, as stereotypes and bias are inherently ambiguous and subjective concepts. As reflected in the dataset annotation scheme, even human annotators may disagree on the presence and intensity of a gender stereotype: depending on personal background, values, and experiences, the same expression may be seen as clearly stereotypical, neutral, or even positive [2, 5]. LLMs, in turn, are highly sensitive to prompting strategies, a property that is also exploited in adversarial prompting scenarios [6]. Without a clear operational definition, an LLM acting as a generalizer over its training data may therefore struggle to align with the intended task interpretation. By explicitly grounding the model in the task-specific definition, we steer it toward the intended linguistic and conceptual framing of gender stereotypes, promoting more consistent and accurate judgments.

To address the challenge comprehensively, we develop three systems corresponding to the three challenge tracks: a zero-shot system, a few-shot system, and an encoder-only fine-tuned system. All three systems are applied to the jointly treated problem of Gender Stereotype Detection and Classification. Section 2 describes the architecture and design choices of each proposed system. Section 3 analyzes the experimental results, first reporting the leaderboard outcomes on the test set based on our submissions and then presenting the internal experimentation conducted to obtain those submissions. Finally, Section 4 concludes the report by summarizing our observations and discussing implications and potential directions for improvement.

2. Description of the System

This section describes the systems developed for the GSI:detect challenge and the design choices underlying their implementation. In particular, we detail three systems corresponding to the zero-shot, few-shot, and encoder-only fine-tuned settings, highlighting their shared components as well as the differences introduced by prompting strategies, in-context examples, retrieval mechanisms, and supervised training. The goal of this section is to provide a clear and systematic account of how each system operationalizes the task definitions and how these design decisions support the experimental analysis presented in the subsequent section.

2.1. Zero-Shot System

Zero-shot learning [7] is typically defined as the ability to generalize to previously unseen classes by transferring knowledge from seen classes, often through auxiliary semantic information. In the context of prompting Large Language Models, the notion of *zero-shot* is used in a more operational sense: the model is queried without task-specific in-context examples of solved instances (shots), which would instead constitute *few-shot* prompting [8], and without any parameter updates, and is expected to perform the task based solely on natural-language instructions. Under this constraint, the

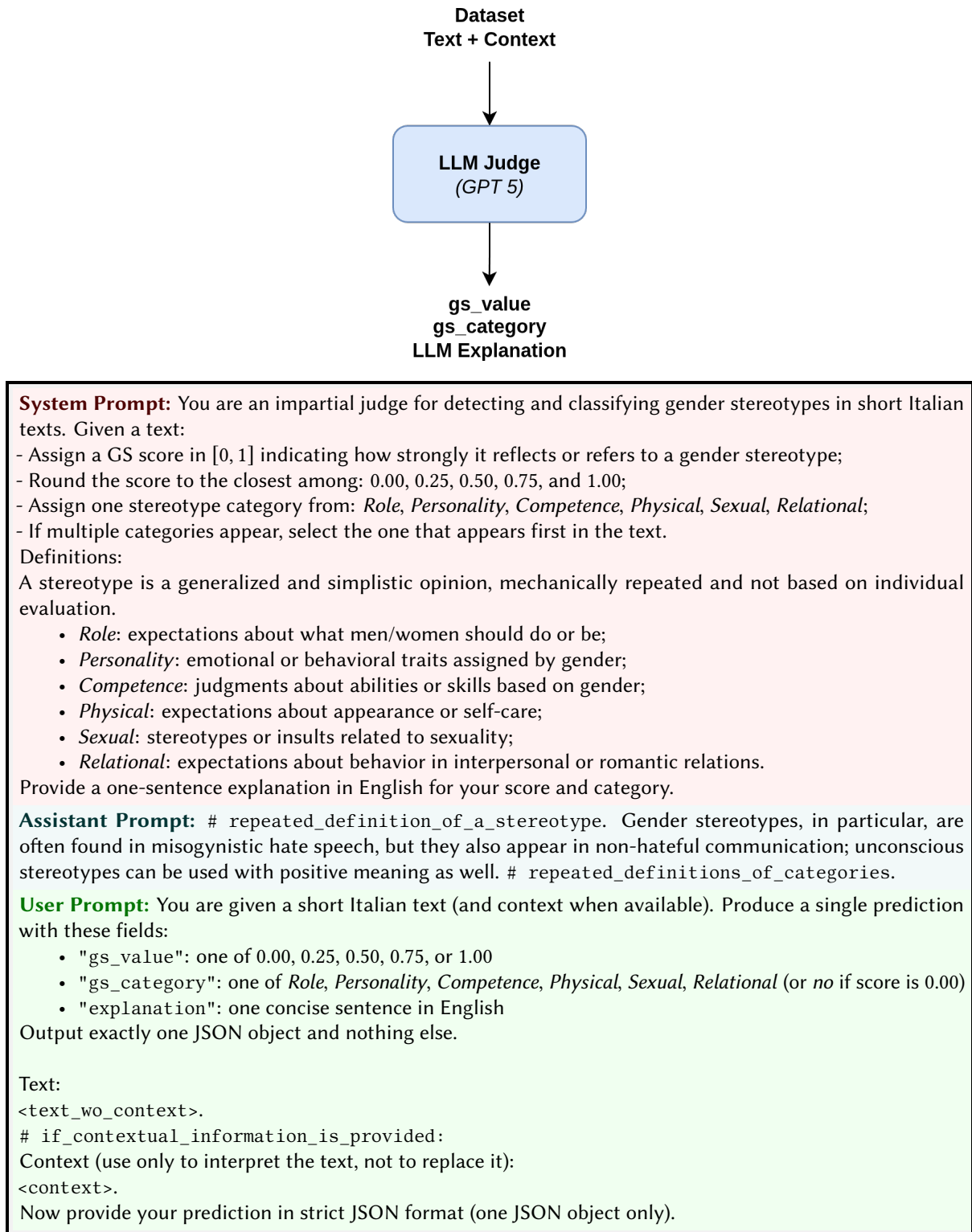


Figure 1: Architecture of the zero-shot system and prompt template with dynamically replaced placeholders.

prompt becomes the main mechanism for steering the model toward the intended behavior. As a result, performance can vary substantially with the formulation of the instructions, the degree of specificity, and the linguistic patterns used to express task definitions and output constraints.

The architecture of our zero-shot system is linear, as illustrated in Figure 1. In both zero-shot and few-shot experiments, we employ GPT-5 as an LLM-as-a-Judge to evaluate the input texts, leveraging

its strong instruction-following and reasoning capabilities as the most recent model in the GPT family. The system is controlled through a three-level prompting scheme consisting of a system prompt, an assistant prompt, and a user prompt.

The system prompt assigns the role of an impartial judge, specifies the task requirements, and introduces the stereotype categories. The assistant prompt then provides the operational definition of gender stereotypes and reiterates the category definitions to reinforce the shared task framing (reiterations are omitted in Figure 1). Finally, the user prompt supplies the instance-level input and constraints the expected output format, requiring the fields `gs_value`, `gs_category`, and `explanation`, defined as “a one-sentence explanation in English” and denoted as “LLM Explanation” in the scheme in Figure 1. It provides the input text, separated from an optional description of its surrounding context when available. The context field is used exclusively to separate contextual information from the text: the model is first presented with the text alone and is then explicitly instructed to interpret it within the given context, using the formulation: “Context (use only to interpret the text, not to replace it)”. A lightweight preprocessing step is applied beforehand to separate the text from its context.

Requiring the model to generate a one-sentence explanation is intended to encourage more accurate responses by implicitly eliciting reasoning during inference [9]. Since explanations are not part of the official submission format, this field is discarded after generation.¹ As the two tasks are treated jointly, both `gs_value` and `gs_category` are predicted within a single model call.

2.2. Few-Shot System

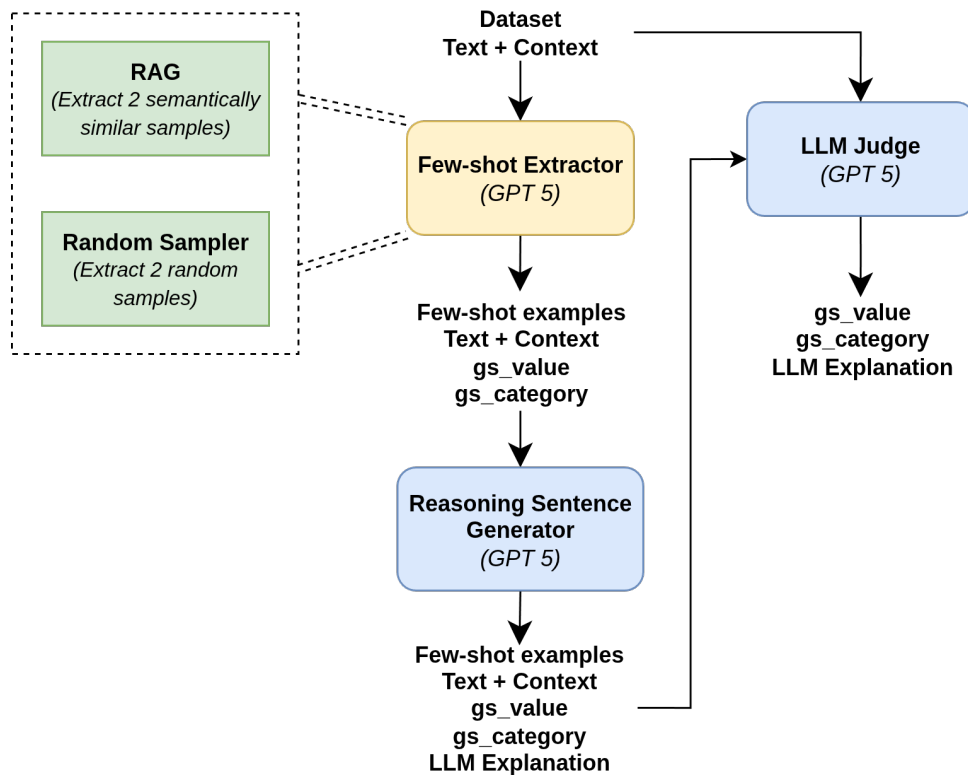


Figure 2: Architecture of the few-shot system.

In-context learning can often yield surprisingly strong results, motivating us to explore a few-shot variant of our approach, shown in Figure 2.

In this setting, requesting an explanation in the user prompt while providing few-shot examples that omit the explanation field would create a mismatch between the demonstrated output template

¹Code and supplementary materials, including the model-generated explanations, are available at: [GSI-detect_DIAG_Sapienza](https://github.com/GSI-detect_DIAG_Sapienza).

and the required output format. Since in-context learning relies on pattern induction from the provided examples [8], such structural inconsistencies can reduce the model’s ability to reliably follow instructions and reproduce the intended schema. To maintain alignment between demonstrations and the target output format, we therefore ensure that they contain the same fields.

For experiments that include explanations, we adopt a two-step procedure. First, in a separate GPT-5 call, we generate one-sentence explanations for the ground-truth examples by providing the text, context, and the gold `gs_value` and `gs_category`, and prompting the model to justify these gold labels. We then augment the few-shot demonstration JSON objects with the resulting explanation field before using them for few-shot prompting. This setup uses two GPT-5 calls: one to generate explanations for demonstrations and one to act as a judge at inference time, as in the zero-shot setting.

To support a more informative selection of few-shot examples, we implement a Retrieval-Augmented Generation (RAG) that selects part of the examples based on textual similarity to the input instance and complements them with additional examples sampled at random. This design exposes the model to both similar and diverse patterns and has been shown to improve few-shot performance [10]. We use LanceDB² to build the retrieval database. In LanceDB, we configure an embedding function from the registry and include it in the table schema. The embedding function automatically generates embeddings when data is inserted into the table and similarly embeds the queried text at retrieval time. We then retrieve the most similar examples by comparing the query embedding to the stored text embeddings in the database. LanceDB supports the most popular embedding providers. For sentence-transformers, the default model is `all-MiniLM-L6-v2`.

2.3. Encoder-Only Fine-Tuned System

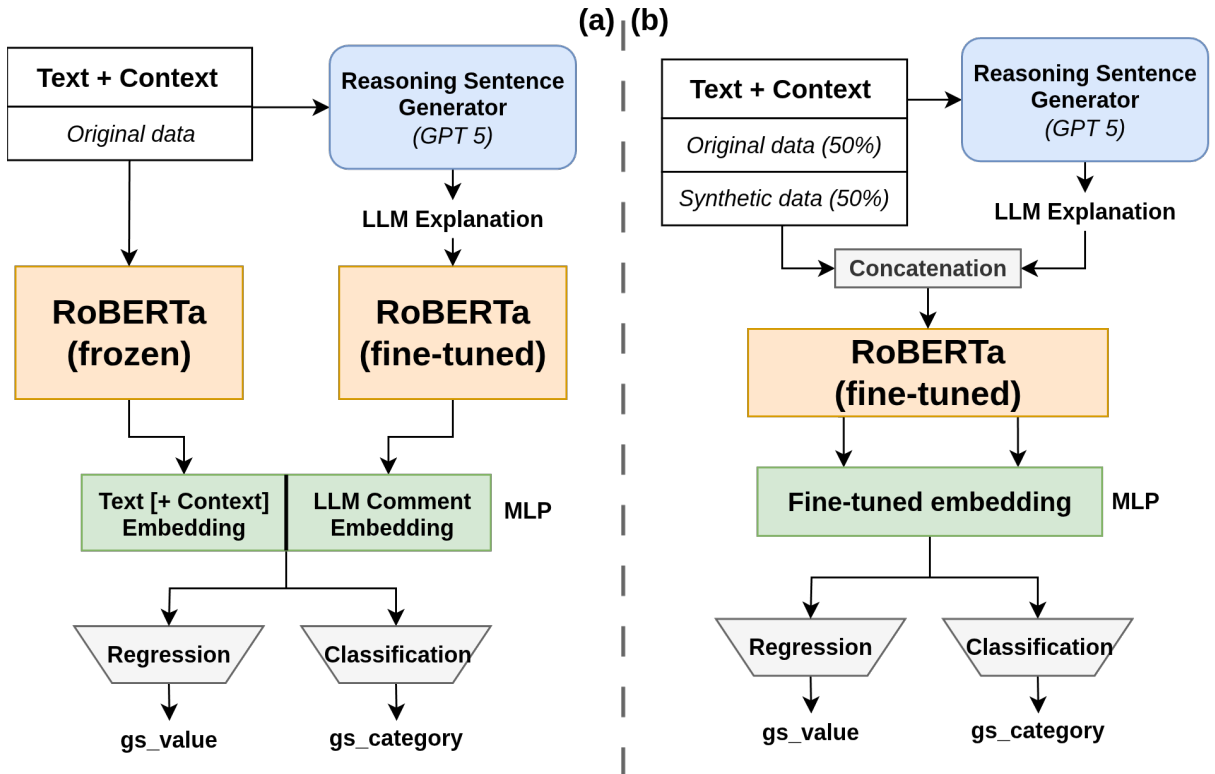


Figure 3: Architecture of the encoder-only fine-tuned system. Panel (a) illustrates a preliminary dual encoder configuration evaluated during ablation studies (Section 3.2.2) that yielded suboptimal performance. Panel (b) shows the final submitted single encoder architecture adopted for the encoder-only track.

²<https://lancedb.com/>

Before the advent of the LLM-as-a-Judge paradigm, the dominant approach for evaluating language systems relied on supervised fine-tuning over task-specific datasets [11]. Nowadays, few-shot or zero-shot inference can yield competitive performance, while in other cases, fine-tuning remains more effective for eliciting specialized behaviors from pretrained models. The relative effectiveness of zero-shot prompting, few-shot in-context learning, and supervised fine-tuning is highly task-dependent and remains an open question, particularly for nuanced phenomena such as gender bias and stereotype detection. For this reason, in addition to zero-shot and few-shot approaches, we also explore the third submission track of the GSI:detect challenge and develop a fine-tuned system.

The architecture of our fine-tuned system belongs to the class of encoder-only models, which focus on learning discriminative representations. The system, illustrated in Figure 3, takes as input the original Italian texts from the dataset, with optional context concatenated to the textual sample when available. In parallel, reasoning sentences corresponding to the explanation field are generated through a separate call to a GPT-5-mini model. In this preliminary step, the LLM is prompted to assess the presence of gender stereotypes and to perform the joint regression and classification task. As a result, the overall architecture comprises two components: a GPT-5-mini-based Reasoning Sentence Generator and a RoBERTa-based judge model that is fine-tuned to perform the final predictions.

2.3.1. RoBERTa-Based Judge

The RoBERTa-based judge [12] employs a single encoder that takes as input the concatenation of the text, the (optional) context, and the LLM-generated explanation. Fine-tuning is performed directly on this encoder, producing a feature representation in the model’s embedding space. The architecture is completed by two task-specific heads, one for regression and one for classification, which are trained jointly using a weighted loss function to balance the two objectives.

2.3.2. Data Augmentation

Before fine-tuning, data augmentation is applied to mitigate overfitting, which can arise due to the limited size of the original dataset. Additional samples are generated through a separate GPT-5 call, in which the model is tasked with producing new Italian texts that express the same gender stereotype category and value as the original samples. For each instance in the original dataset, a new text is generated with the same `gs_value` and `gs_category` but with different wording, structure, and contextual framing.

The augmentation process mirrors the prompting strategy used in the zero-shot system and relies on a system prompt, an assistant prompt, and a user prompt. In the system prompt, GPT-5 is assigned the role of an expert linguist specializing in the detection and classification of gender stereotypes in Italian and is instructed to generate realistic and grammatically correct texts. The prompt specifies several constraints: the generated texts must preserve the stereotype intensity level, expressed as `gs_value` (0.00=no stereotype, 0.25=mild, 0.50=moderate, 0.75=strong, and 1.00=explicit/severe), maintain the original `gs_category`, and remain natural and plausible while differing substantially from the source text. The assistant prompt reiterates the definitions of gender stereotypes and their categories to reinforce adherence to these constraints. Finally, the user prompt provides the original text, `gs_value`, and `gs_category` and requests the generation of new text matching these attributes. The full prompt template is reported in Appendix A.1.

To prevent the inclusion of incoherent outputs, hallucinations, or violations of the imposed constraints, all generated samples undergo manual curation and filtering. As a result, 195 data samples are added to the development dataset, substantially increasing its diversity. An example of an original development sample and its generated counterpart is shown below; the English translation is provided in Appendix A.2.

Original GSI Dev Sample (ID: GS0458)

text_wo_context: “IO, mammta e tu, non ne posso più ecc ecc ecc. Dobbiamo anche dire, che i matrimoni duravano una vita e non si sentivano tutti i giorni come oggi di femminicidi, si scherza MOLTO sui sentimenti delle persone, perche’ la torta viene mangiata prima della festa e datosi che non c’è più vergogna di NIENTE, molto facilmente si dice che non era buona e non c’è più desiderio, ma sé si mangia un poco alla volta, la torta è desiderata e buona, poche parole a buon intenditore!”

context: “Commento a una vecchia foto di un matrimonio in cui sia gli sposi, sia gli invitati sono tutti molto seri, con la dicitura: *Un matrimonio d’altri tempi.*”

gs_value: 0.25

gs_category: sexual

Augmented Counterpart (Synthetic Sample)

text: “Quando ascolto mia nonna parlare dei matrimoni di una volta, ripete che le ragazze perbene sapevano aspettare e non si concedevano subito; forse è anche per questo che le storie duravano.”

gs_value: 0.25

gs_category: sexual

3. Results

In this section, we report both the leaderboard results obtained with the official GSI:detect evaluator on the test set and the findings from our internal experimentation conducted before submission.

3.1. Leaderboard Results

Our submission consists of six evaluation runs on the official test set: one run for the zero-shot track, one for the few-shot track, and four runs for the encoder-only fine-tuning track. For the encoder-only fine-tuned system, the four runs correspond to two different experimental configurations. Specifically, Runs 1 and 2 use only the original dataset texts and generated reasoning sentences, whereas Runs 3 and 4 use the augmented version of the dataset. Each configuration includes two runs because we consider two alternative formats for the regression output. In Run 1 (and analogously Run 4), the model predicts a continuous `gs_value` in the full range $[0, 1]$. In Run 2 (and analogously Run 3), the predicted `gs_value` is rounded to the nearest value in the discrete set $\{0.00, 0.25, 0.50, 0.75, 1.00\}$, reflecting the granularity of the ground-truth annotation scheme.

Table 1 and Table 2 summarize the official leaderboard results for the main task and the sub-task, respectively. We first discuss the main task of *Gender Stereotype Detection*. Here, our zero-shot and few-shot systems achieve the first and second positions on the leaderboard, respectively, outperforming all other submitted systems. The official evaluated metric for this task is: $1 + \frac{1}{NMSE}$, where *NMSE* denotes the Normalized Mean Square Error, with higher values indicating better performance. In contrast, the encoder-only fine-tuned system yields substantially lower scores, suggesting that further refinement of this approach is required.

Table 2 reports the results for the sub-task of *Gender Stereotype Classification*. In this setting, the few-shot system achieves the highest ranking among our submissions, placing 14th overall. The primary evaluation metric is the Micro-averaged F1 score, with higher values corresponding to better performance. Compared to the results on the main task, these findings support our hypothesis that the joint treatment of detection and classification is particularly beneficial for the regression task, while further improvements are necessary to strengthen performance on the classification sub-task.

Table 1

Main Task (Regression) results achieved on the test dataset.

Track	Run	Ranking	1+1/NMSE	NMSE	MSE
zero-shot	best	1	0.700	0.4292	0.0766
few-shot	best	2	0.685	0.4594	0.0820
encoder-only	1	45	0.470	1.1260	0.3752
encoder-only	2	46	0.462	1.1644	0.3691
encoder-only	4	47	0.453	1.2072	0.3562
encoder-only	3	48	0.446	1.2409	0.3601

Table 2

Sub-Task (Classification) results achieved on the test dataset.

Track	Run	Ranking	F1 Micro	F1 Macro
few-shot	best	14	0.606	0.55
zero-shot	best	19	0.581	0.52
encoder-only	1	32	0.375	0.29
encoder-only	2	33	0.357	0.29
encoder-only	3	34	0.349	0.29
encoder-only	4	35	0.349	0.29

Table 3

Ablation study of the few-shot system from Figure 2. Evaluation results achieved during experimentation on the development dataset.

System	Architecture	MSE	Accuracy	F1 Micro
few-shot	full	0.0765	0.5243	0.6699
few-shot	w/o reasoning sentence	0.0781	0.56	0.6250
few-shot	w/o RAG database	0.070	0.600	0.6350

3.2. Internal Experimentation

Our internal experimentation is structured into two ablation studies, focusing respectively on the few-shot system and the encoder-only fine-tuned system.

3.2.1. Ablation Study of the Few-Shot System

In few-shot experiments, we conduct two ablation studies: 1) with vs. without reasoning sentences and 2) with vs. without the RAG pipeline. In the first ablation, we disable the Reasoning Sentence Generator in Figure 2: we neither generate reasoning sentences for the few-shot examples nor request them in the user prompt. In the second ablation, we remove retrieval and sample all four few-shot examples uniformly at random from the development set. To avoid overlap between examples and the query instance, we apply a partitioning of the development data at each iteration and sample demonstrations only from the partition that excludes the query text.

Table 3 presents evaluation results on the development dataset, where gold labels are available. Since our approach treats the two tasks jointly, we report both Mean Squared Error (MSE) for the regression task (non-normalized; lower is better) and F1 Micro for the classification task. For the regression task, we additionally calculate accuracy.

Based on these results, we select the configuration without RAG to produce the test-set submission, denoted as the best run in Table 1 and Table 2. This configuration yields stronger performance on the regression task, whereas the full architecture achieves the best classification performance. However, the retrieval of RAG examples takes time. Finally, across settings, including reasoning sentences consistently improves performance, even though the gains are modest.

3.2.2. Ablation Study of the Encoder-Only Fine-Tuned System

Model selection was performed by training the model on 70% of the original dataset and evaluating predictions on the remaining 30%.

To address the limited size of the training data and mitigate the risk of model collapse, we adopted an encoder-only architecture with a single-encoder concatenation. In this design, both the text and the LLM-generated explanation are processed by a single encoder instance, resulting in approximately 125M trainable parameters and reducing overfitting risk. The model features a shared Multi-Layer Perceptron (MLP) with a hidden dimension of 768, which feeds into a classification head with hidden dimension 768 and a regression head with hidden dimension 512.

We employed a robust training strategy to stabilize learning on the small training dataset (322 samples). The dataset exhibited a strong class imbalance: to mitigate this problem, we used inverse-frequency class weights in the loss function. Training stability was further improved through gradient clipping ($\text{max_norm}=1.0$) and label smoothing with a factor of 0.1. A dropout rate of 0.3 was applied throughout the network: after each ReLU activation in the shared MLP and both before the hidden layer and before the output layer of each task-specific head. The classification head was initialized using Xavier uniform weights (gain 0.1) to ensure a neutral starting point. Optimization was performed with AdamW using a learning rate of $2e^{-5}$, a batch size of 4, and weight decay of 0.01. Training was run for up to 150 epochs with early stopping (patience 15).

Our experimentation followed an iterative ablation process, starting from a simple baseline and progressively increasing the architecture complexity until reaching the most promising configuration. The evolution of this process is detailed below and summarized in Table 4.

Baseline: Single Encoder We initially established a baseline using a single encoder architecture, in which the input text and the corresponding LLM-generated reasoning sentence are concatenated and processed by a RoBERTa-based encoder ($\sim 125M$ parameters). We applied a dropout rate of 0.5 to reduce overfitting over the training set. Due to the simplicity of the task, training was performed for up to 50 epochs, without data augmentation. Despite avoiding complete collapse, the model suffered from slow convergence and suboptimal performance, achieving a best validation Micro F1 score of ≈ 0.13 on the classification sub-task. This result indicates that the architecture struggled to effectively exploit the additional contextual information provided by the reasoning sentences.

Dual Encoder To improve upon the baseline, we hypothesized that a dual encoder architecture could better capture the distinct features of the original texts and the explanations. This setup employed two independent RoBERTa encoders ($\sim 250M$ parameters in total), whose outputs were concatenated and processed by a shared MLP. The first encoder, receiving the text and context, was kept frozen, while the second encoder, processing the reasoning sentence, was fine-tuned for up to 50 epochs. A dropout rate of 0.5 was applied. However, this configuration resulted in model collapse: the significant increase in parameters (+100%), combined with the low sample count of the training data, led to immediate overfitting. The model consistently collapsed to predicting a single class and achieving a low best F1 score of ≈ 0.04 , regardless of hyperparameter adjustments. Supposedly, the reason behind the model collapse was the excessive model complexity with respect to the available dataset size.

Optimized Single Encoder (Final Model) The final delivered architecture reverts to the single encoder architecture and focuses on optimization and regularization to maximize stability and performance. Compared to the baseline, dropout was reduced, gradient clipping and label smoothing were introduced. Crucially, the training set was augmented in two steps with synthetic data, generated as previously described. A first experiment was conducted with a training set with a 20% of synthetic samples. A second experiment was conducted with 50% synthetic samples. Training was extended to 150 epochs with early stopping. This configuration yielded the best results, achieving high stability and improved performance, with no signs of model collapse. No further experimentation was conducted.

Table 4

Ablation study of model architectures and configurations. The dual encoder scaling attempt failed due to data scarcity, while the optimized single encoder with synthetic data augmentation achieved the best performance.

Experiment	Architecture	Params	Augmentation	Dropout	Epochs	Best Val F1
Baseline	Single Enc.	~125M	None	0.5	50	0.1257
Scaling Attempt	Dual Enc.	~250M	None	0.5	50	0.0408
Optimized	Single Enc.	~125M	Synthetic (20%)	0.3	100	0.8293
Final	Single Enc.	~125M	Synthetic (50%)	0.3	150	0.8834

In our experimentation, we adopted a set of focused design choices aimed at validating a coherent modeling hypothesis within the constraints of the shared task setting. While these choices proved effective, they also naturally delimit the scope of the experimental analysis and suggest several directions for further investigation.

4. Discussion and Conclusions

In our experimentation, we adopted a set of focused design choices. While these choices proved mostly effective, they naturally suggest several directions for further investigation.

First, we exclusively considered a joint formulation of *Gender Stereotype Detection* and *Gender Stereotype Classification*. Although treating the two tasks independently could offer additional insights, our results indicate that the joint approach is particularly effective for the regression task. A systematic comparison with fully independent task formulations would nonetheless be valuable to quantify the performance margin introduced by joint modeling and to further characterize task interactions.

Our systems rely on GPT-5 as the core evaluator model. While this choice is justified by its strong reasoning capabilities and overall performance, we did not conduct a comparative evaluation across different LLM families or providers. A broader cross-model analysis could help determine whether similar gains can be achieved with alternative architectures or whether the observed performance is tied to specific properties of GPT-5.

Another deliberate design choice concerns prompt structuring. We extended the standard system-prompt configuration by introducing an assistant prompt that reiterates key definitions. This is motivated by the hypothesis that reinforcing background knowledge can improve model alignment with the task definition. A controlled ablation removing the assistant prompt would allow for an assessment of its individual contribution.

Regarding the zero-shot system, we did not perform a dedicated ablation study to isolate the effect of enforcing explanation generation. Such an analysis could further clarify the role of implicit reasoning in improving prediction quality.

In the few-shot setting, we expected the integration of Retrieval-Augmented Generation to yield improvements, but did not observe it. A possible explanation is the limited semantic overlap between development and test samples, which reduces the benefit of retrieving highly similar examples. In addition, the remaining two examples are sampled randomly rather than selected to maximize contrast, and the small number of demonstrations limits coverage of all stereotype categories in the classification task. Exploring alternative retrieval strategies, varying the number of examples, or enforcing class-balanced selection could help address these factors. Although the few-shot architecture with RAG achieved stronger classification performance on the development set, we ultimately submitted the configuration without RAG due to its better regression performance and lower computational overhead. As a result, the potential leaderboard impact of the RAG-enhanced few-shot system on the classification task remains an open question.

Finally, it is noteworthy that zero-shot and few-shot systems outperform the fine-tuned model on the leaderboard. While this outcome may appear counterintuitive, fine-tuning performance is highly sensitive to architectural choices, backbone selection, and data availability. Our ablation study indicates

that encoder-only fine-tuning remains promising, but is strongly constrained by the limited size and imbalance of the training data. Further refinements, alternative architectures, and training on larger and more balanced datasets could substantially improve performance. Additionally, more systematic experimentation with different levels of data augmentation would help clarify its impact.

Overall, our zero-shot and few-shot systems demonstrate strong performance and robustness in low-data settings, validating the effectiveness of joint modeling and structured prompting. The encoder-only fine-tuned system represents an avenue for future improvement and extension.

5. Acknowledgments

This work has been carried out while Olga Sorokoletova and Emanuele Musumeci were enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome. We acknowledge partial financial support from PNRR MUR project PE0000013-FAIR.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-5.2 and the Grammarly plugin in order to: Paraphrase and reword; Improve writing style; and Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication’s content.

References

- [1] G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini, Gsi: detect at evalita 2026: Overview of the task on detecting gender stereotypes in italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] A. Testolin, Can neural networks do arithmetic? a survey on the elementary numerical skills of state-of-the-art deep learning models, *Applied Sciences* 14 (2024) 744.
- [4] S. Cavagnoli, F. Dragotto, et al., *Sessismo*, Mondadori Education, 2021.
- [5] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, M. A. Stranisci, An italian twitter corpus of hate speech against immigrants, in: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018), 2018.
- [6] Y. Zeng, et al., How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms, 2024. URL: <https://arxiv.org/abs/2401.06373>. arXiv:2401.06373.
- [7] Y. Xian, B. Schiele, Z. Akata, Zero-shot learning-the good, the bad and the ugly, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4582–4591.
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, *Advances in neural information processing systems* 33 (2020) 1877–1901.
- [9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems* 35 (2022) 24824–24837.
- [10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, *Advances in neural information processing systems* 33 (2020) 9459–9474.

- [11] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146 (2018).
- [12] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: <https://arxiv.org/abs/1907.11692>. arXiv:1907.11692.

A. Appendix

A.1. Prompt Template for Data Augmentation

Below, we report the prompt template adopted during the data augmentation procedure. Placeholders enclosed in angle brackets (e.g., <text_wo_context>) are dynamically replaced with instance-specific values.

System Prompt: You are an expert linguist specializing in detecting and classifying gender stereotypes in Italian texts. Your task is to augment a dataset by generating new realistic Italian texts that contain gender stereotypes with the SAME stereotype score and category as the original. The generated text must:

- Be natural, grammatically correct Italian;
- Preserve the stereotype intensity level (score) of the original;
- Use different wording and context while maintaining the stereotype category;
- Sound authentic and plausible.

Stereotype Definitions:

- *Role*: expectations about what men/women should do or be;
- *Personality*: emotional or behavioral traits assigned by gender;
- *Competence*: judgments about abilities or skills based on gender;
- *Physical*: expectations about appearance or self-care;
- *Sexual*: stereotypes or insults related to sexuality;
- *Relational*: expectations about behavior in interpersonal or romantic relations.

Score intensity:

0.00=no stereotype; 0.25=mild; 0.50=moderate; 0.75=strong; 1.00=explicit/severe.

Assistant Prompt: # repeated_definition_of_a_stereotype. Gender stereotypes, in particular, are often found in misogynistic hate speech, but they also appear in non-hateful communication; unconscious stereotypes can be used with positive meaning as well. # repeated_definitions_of_categories.

User Prompt: Generate a NEW Italian text with the SAME gender stereotype as the original.

IMPORTANT CONSTRAINTS:

- Create a completely different sentence (different wording, structure, scenario)
- Preserve the exact stereotype category and intensity score
- Output ONLY a valid JSON object with NO additional text

Original text:

<text_wo_context>

Stereotype category:

<gs_category>

Stereotype intensity (0.00 – 1.00):

<gs_value>

if_contextual_information_is_provided:

Context (for interpretation only):

<context>

Respond with ONLY the following JSON structure:

```
{
  "text": "new Italian sentence here",
  "gs_value": <gs_value>,
  "gs_category": <gs_category>
}
```

A.2. English Translation of Data Augmentation Example (ID: GS0458)

The translations are provided for illustrative purposes only and do not affect the evaluation, which is conducted on the Italian texts.

Original GSI Dev Sample (ID: GS0458)

text_wo_context: "Me, mom, and you, I can't take it anymore etc. etc. etc. We also have to say that marriages used to last a lifetime, and you didn't hear every day like today about femicides. People joke A LOT about other people's feelings, because the cake is eaten before the party, and since there is no longer shame about ANYTHING, it is very easy to say it wasn't good and there is no longer desire. But if you eat it little by little, the cake is desired and good, a few words to the wise!"

context: "Comment on an old wedding photo in which both the bride and groom and the guests appear very serious, with the caption: *A wedding from another era.*"

gs_value: 0.25

gs_category: sexual

Augmented Counterpart (Synthetic Sample)

text: "When I hear my grandmother talk about weddings in the past, she repeats that respectable girls knew how to wait and did not give themselves away immediately; maybe that is also why relationships lasted."

gs_value: 0.25

gs_category: sexual

Festa at GSI:detect: In-Context Learning vs. Fine-Tuning for Gender Stereotype Detection

Donato Festa

University of Bari Aldo Moro, Italy

Abstract

This paper describes the system developed for the GSI:detect task at EVALITA 2026. The goal of the task is to quantify the intensity of gender stereotypes in Italian text (Main Task) and classify them into predefined categories (Subtask). We propose a comparative study utilizing **Google Gemini 2.5 Flash**, contrasting In-Context Learning strategies (Zero-Shot and Few-Shot) against a Fine-Tuning approach based on **UmBERTo**. Our results reveal a significant trade-off between classification performance and intensity estimation. The Few-Shot approach, enhanced by an ensemble strategy, achieved better performance in stereotype categorization (Micro F1: 0.6708), demonstrating superior capability in detecting implicit biases, particularly in the “Relational” category. Conversely, the best Zero-Shot configuration proved more effective for intensity regression (NMSE Score: 0.6205). This paper details the prompt engineering strategies, the experimental setup, and provides an error analysis of the different configurations.

Keywords

Gender Stereotype Detection, Large Language Models, Few-Shot Learning, Prompt Engineering, Ensemble Learning

1. Introduction

Gender stereotyping in language is a widespread issue that reinforces social inequalities and harmful biases. The GSI:detect task [1], organized within the EVALITA 2026 campaign [2], addresses this challenge by proposing a comprehensive evaluation of stereotypes on Italian texts. The task involves estimating the intensity of stereotypes on a continuous scale (Main Task) and classifying them into predefined categories (Subtask).

In this work, we place a particular emphasis on the categorization of gender stereotypes (the Subtask), as the identification of specific bias types is critical for developing targeted mitigation strategies and understanding the semantic nuances of various stereotype forms. Understanding whether a sentence conveys a stereotype related to competence rather than physical appearance, for instance, provides a more granular and actionable analysis of gender bias in social media.

Stereotypes can range from explicit remarks (e.g., regarding physical appearance) to subtle and implicit associations (e.g., questioning professional competence based on gender), making automatic detection a complex semantic challenge.

In this work, we explore the capabilities of Large Language Models (LLMs), specifically **Google Gemini 2.5 Flash**, comparing In-Context Learning strategies (Zero-Shot and Few-Shot) against a standard Fine-Tuning approach based on the **UmBERTo** architecture. Our contribution can be summarized as follows:

1. We conduct a systematic comparison of three distinct approaches for Italian stereotype detection:
 - **Zero-shot** prompting, where the model is given only the task description without examples.
 - **Few-shot** In-Context Learning, where the model is provided with a few annotated examples to guide its predictions.
 - **Fine-Tuning**, where we adapt a pre-trained UmBERTo model to the specific task using the provided dataset.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ d.festa5@studenti.uniba.it (D. Festa)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. We investigate the impact of context availability by experimenting with different numbers of examples ($k \in \{6, 10, 12, 14\}$) in the Few-Shot setting to determine the optimal trade-off between prompt complexity and accuracy.
3. For each approach we propose (at least) an ensemble strategy that aggregates predictions from multiple runs to improve classification stability and reduce variance in intensity regression.

2. Task and Dataset

The GSI:detect challenge targets the identification of gender stereotypes in Italian texts. It is divided into two parts:

- **Main Task (Intensity Prediction):** A regression task where the system must predict a scalar value ('gs_value') representing the degree of stereotyping, ranging from 0 (no stereotype) to 1 (extreme stereotype).
- **Subtask (Stereotype Categorization):** A multi-class classification task where the system must assign a specific label to a sentence. The categories include: *Physical*, *Sexual*, *Role*, *Relational*, *Competence*, and *Personality*. If no stereotype is present, the label is *No*.

2.1. Dataset

The dataset provided by the organizers consists of User-Generated Content (UGC) collected from online sources (e.g., social media comments), characterized by informal language, emojis, and non-standard grammar. The data is split into:

- A **Development Set of 200 sentences**, which we utilized for training and validation (via Cross-Validation) in the Fine-Tuning approach, and as the source of examples for the Few-Shot prompting.
- A **Test Set of 810 sentences**, used for the final evaluation.

2.2. Data Preprocessing and Augmentation

Given the limited size of the development set and the natural class imbalance it exhibits, we implemented a custom **Conservative Data Augmentation** pipeline to mitigate this during the Fine-Tuning phase. Unlike aggressive augmentation techniques that might alter the semantic meaning (e.g., back-translation), our approach utilizes three safe transformations:

1. **Synonym Replacement:** Substituting a single adjective or noun with a contextually appropriate synonym (e.g., “*donna*” → “*signora*”).
2. **Safe Insertion:** Injecting neutral adverbs (e.g., “*sicuramente*”, “*ovviamente*”) at the beginning or end of the sentence.
3. **Pattern Paraphrasing:** Rewriting common syntactic patterns (e.g., “*sono sempre*” → “*risultano sempre*”).

We oversampled the minority classes to reach a target of 60 samples per class. For In-Context Learning, no augmentation was applied.

2.3. Evaluation Metrics

The official evaluation metrics are:

- **NMSE Score:** The official ranking metric for the Main Task. It is derived from the Normalized Mean Squared Error as $1/(1 + NMSE)$, providing a score between 0 and 1 where higher is better.
- **Micro F1-Score:** The official ranking metric for the Subtask (Classification). It evaluates classification performance globally by aggregating true positives, false negatives, and false positives across all samples, effectively giving equal weight to each *instance*.

- **Macro F1-Score:** In addition to the official Micro F1-Score we also used Macro F1, which calculates the arithmetic mean of the per-class F1 scores. This metric treats all classes equally regardless of their frequency, highlighting the model’s ability to detect minority stereotype categories.

3. System Description

To address the GSI:detect challenge, we developed a system that contrasts Fine-Tuning with In-Context Learning (ICL) strategies utilizing Large Language Models. We focused on Google Gemini 2.5 Flash as our generative engine due to its strong reasoning capabilities and efficiency.

3.1. Fine-Tuning Approach (UmBERTo)

We employed **UmBERTo** [3] (Musixmatch/umberto-commoncrawl-cased-v1), a RoBERTa-based model pre-trained on a large corpus of Italian CommonCrawl data. Instead of a simple linear probe, we designed a custom **Multi-Layer Perceptron (MLP) head** to process the ‘[CLS]’ token embedding. The head consists of a Dropout layer ($p = 0.1$), a dense projection to 128 units with Tanh activation, a second Dropout, and a final linear layer.

We fine-tuned two separate models:

- **Regression Model (Main Task):** The final layer is followed by a **Sigmoid** activation function to constrain the output strictly within the $[0, 1]$ range. We optimized the model using *Mean Squared Error* (MSE) loss.
- **Classification Model (Subtask):** The final layer returns the raw logits for the 7 target classes. We optimized the model using *Cross-Entropy Loss* (weighted to handle residual imbalance).

Training was performed for 15 epochs (Batch Size=16, LR= $3e^{-5}$, AdamW optimizer) using Stratified K-Fold Cross-Validation ($k = 5$).

3.1.1. Experimental Configurations (Runs).

To explore the trade-offs between stability and specialization, we submitted 5 distinct variations based on the stratified k-fold models. Note that since the *No* label was not permitted in the final submission, all runs mapped non-stereotype predictions to valid categories, but with different strategies:

- **Run 1 (Standard Ensemble):** A weighted ensemble of all 5 folds where, for each sample, the system selects the **valid** category with the highest aggregated probability score.
- **Run 2 (Mean Ensemble):** An unweighted ensemble using simple arithmetic averaging (intensity) and majority voting (categories) among valid classes.
- **Run 3 (Best Single Model):** Inference using only the single model (Fold 2) that achieved the highest validation scores.
- **Run 4 (Single Fold):** A run using only the second best model (Fold 3).
- **Run 5 (Role Fallback):** A weighted ensemble (Run 1) with a deterministic heuristic: instead of selecting the next most probable class, whenever the model predicts *No*, the system **always** defaults to *Role* (the most frequent category).

3.2. In-Context Learning (Gemini)

We leveraged Gemini 2.5 Flash [4] via API with generation temperature set to 0 for reproducibility. We disabled safety filters (BLOCK_NONE) to process the potentially offensive content of the dataset.

3.2.1. Prompt Engineering Strategies

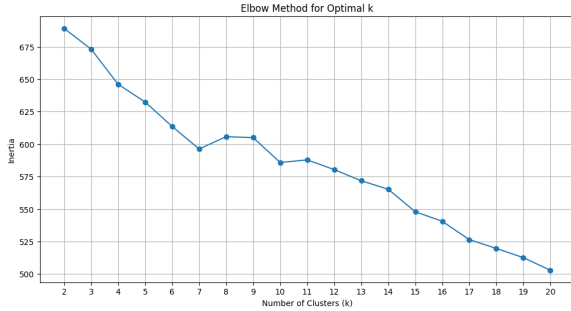
We adopted an iterative Prompt Engineering approach, testing distinct strategies for both Zero-Shot and Few-Shot settings.

Zero-Shot Iterations. Our experimentation began with a baseline template (V1). However, preliminary tests revealed suboptimal performance, characterized by frequent hallucinations and a lack of sensitivity. Consequently, we discarded V1 and evolved our approach into four distinct refined templates (V2–V5):

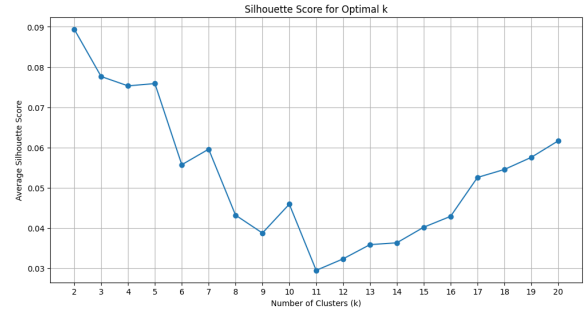
- **False Positive Reduction (V2):** An English prompt explicitly instructing the model to distinguish between factual topics (e.g., cooking) and stereotypes (e.g., gender obligations), aiming to improve precision.
- **Cultural Adaptation (V3):** A prompt entirely in **Italian**, defining categories with culturally specific examples to better align with the target language.
- **Chain-of-Thought (V4 & V5):** We implemented a *Chain-of-Thought* (CoT) mechanism by requiring the model to output a "reasoning" field in the JSON before the final label. This forces the model to articulate its logic (e.g., *"Is this an offensive generalization?"*) before committing to a score. We tested this strategy in both Italian (V4) and English (V5).

Few-Shot Strategies. Instead of randomly selecting examples, we implemented a data-driven selection process to construct the optimal few-shot prompt. We experimented with three selection strategies:

1. **Manual Selection ($k = 6$):** A baseline set of hand-picked examples covering the main categories.
2. **Semantic Clustering ($k = 10, 12$):** We generated sentence embeddings for the entire development set using paraphrase-multilingual-mpnet-base-v2, based on the Sentence-BERT architecture [5]. We then applied K-Means clustering to group semantically similar sentences and selected the samples closest to each cluster centroid. This ensures the model is exposed to the maximum semantic diversity of the dataset.
3. **Hybrid Strategy ($k = 14$):** A "Best-of-Both-Worlds" approach. We combined the optimal clustering set ($k = 12$) with two manually selected "corrective" examples, specifically chosen to address common failure cases (e.g., non-stereotypical sentences often misclassified as stereotypes).



(a) Elbow Method analysis.



(b) Silhouette Score analysis.

Figure 1: Clustering analysis on development set embeddings. The combined view of the Elbow curve and Silhouette scores guided the selection of $k \in \{10, 12\}$ to balance semantic granularity and cluster separation.

3.3. Ensemble Strategy

To mitigate the variance inherent in individual models and maximize robustness, we implemented two distinct ensemble strategies tailored to the nature of each approach.

3.3.1. LLM Ensemble (Zero-Shot & Few-Shot)

For the In-Context Learning approaches, we aggregated the outputs of the four distinct prompt variations (V2–V5 for Zero-Shot; $k \in \{6, 10, 12, 14\}$ for Few-Shot) using an unweighted voting mechanism:

- **Intensity (Main Task):** We computed the simple arithmetic mean of the predicted scores.
- **Categorization (Subtask):** We applied a standard Majority Voting. To resolve ties (e.g., two votes for *Role* and two for *Personality*), we implemented a deterministic tie-breaking rule: the system defaults to the prediction of the highest-performing single run (acting as the “leader” model).

Finally, to strictly adhere to the challenge submission format which disallowed the *No* label for the Subtask, we applied a post-processing step where any residual non-stereotype prediction was mapped to the *Role* category (the most frequent class in the training set).

3.3.2. Fine-Tuning Ensemble

For the fine-tuned UmBERTo models, we adopted a more aggressive aggregation strategy to boost sensitivity:

- **Weighted Voting:** Predictions from the 5 cross-validation folds were aggregated using a weighted voting scheme, where the weight of each model was proportional to its F1-score (for classification) or NMSE (for regression) on the validation set.
- **Forced Role Fallback:** To comply with the final challenge guidelines, which required the prediction of one of the six stereotype categories (excluding the *No* label), for the final run (Run 5) of this approach, we implemented a mandatory fallback mechanism. Whenever the weighted voting resulted in a *No* prediction, the system automatically defaulted to *Role* (the most frequent class in the training distribution) to ensure valid submission outputs.

4. Experimental Setup

To ensure reproducibility, we detail below the computational environment and the hyperparameters used for both the Fine-Tuning and In-Context Learning experiments.

4.1. Computational Environment

All experiments were conducted on a local workstation running **Windows 11**. The hardware configuration included an **Intel Core i7-14700HX** processor, **32GB of RAM**, and an **NVIDIA GeForce RTX 4070** GPU with 8GB of VRAM. We implemented the system using Python, leveraging the PyTorch framework and the Hugging Face Transformers library for the fine-tuning pipeline, and scikit-learn for the clustering operations.

4.2. Hyperparameters and Configuration

- **LLM Generation (Gemini):** For all prompt-based runs, we utilized the `google-generativeai` Python client. We set the generation temperature to 0 to ensure deterministic outputs and disabled safety filters (`HarmBlockThreshold.BLOCK_NONE`) to process the sensitive nature of the dataset without refusals.
- **Fine-Tuning (UmBERTo):** We trained the model for 15 epochs using the **AdamW** optimizer with a learning rate of $3e^{-5}$ and a linear scheduler (10% warmup). To optimize memory usage on the local GPU, we set the **training batch size to 16**. For the ensemble inference phase, we increased the batch size to 32 to maximize throughput.

5. Results

We evaluated the performance of our three approaches (Fine-Tuning, Zero-Shot, and Few-Shot) on the official test set. Table 1 summarizes the results for all 15 experimental runs, reporting the official ranking metric (Micro F1) alongside the internal Macro F1 and NMSE scores.

Table 1

Main results on the GSI:detect Test Set. The table compares the **Micro F1-Score** (Official Ranking Metric for the Subtask), **Macro F1-Score** (Internal Evaluation), and **NMSE Score** (Main Task) across all configurations. The best result for each metric is highlighted in **bold**.

Approach	Configuration (Run)	Micro F1	Macro F1	NMSE Score
Fine-Tuning	Run 3: Single Best Fold (Fold 2)	0.4789	0.4577	0.5409
	Run 4: Single Fold (Fold 3)	0.4525	0.4272	0.5527
	Run 2: Ensemble (Mean)	0.4894	0.4707	0.5604
	Run 5: Ensemble (Forced Role)	0.4683	0.4529	0.5604
	Run 1: Standard Ensemble	0.5211	0.5038	0.5604
Zero-Shot	V2: False Positive Red.	0.5898	0.5515	0.6205
	V3: Italian Cultural	0.5863	0.5427	0.6096
	V4: CoT (Italian)	0.5687	0.5227	0.5548
	V5: CoT (English)	0.6004	0.5605	0.5572
	Ensemble (Voting)	0.5968	0.5520	0.6177
Few-Shot	$k = 6$ (Manual Selection)	0.6356	0.6271	0.4996
	$k = 10$ (Clustering)	0.6496	0.6345	0.5605
	$k = 12$ (Clustering)	0.6567	0.6526	0.5650
	$k = 14$ (Hybrid Strategy)	0.6708	0.6687	0.5500
	Ensemble (Voting)	0.6655	0.6562	0.5717

5.1. Performance Analysis

The results highlight a significant performance gap between In-Context Learning and Fine-Tuning. The Few-Shot approach consistently outperformed all other methods, with the **Hybrid Strategy** ($k = 14$) achieving the best performance among the methods we tested in the official metric (**Micro F1: 0.6708**).

This validates the effectiveness of combining semantic clustering with manually selected corrective examples. Interestingly, while Zero-Shot models generally underperformed in classification, the English Chain-of-Thought configuration (V5) achieved a competitive Micro F1 of 0.6004, suggesting that advanced reasoning prompts can partially compensate for the lack of examples. Fine-Tuning proved to be the weakest approach (best Micro F1: 0.5211), likely due to the limited size of the training set (200 examples) which was insufficient for the model to generalize well on complex categories.

5.2. Error Analysis by Category

To understand the specific strengths and weaknesses of each approach, we compared the per-category F1-scores of the best performing run for each setting. The quantitative breakdown is reported in Table 2, while Figure 2 provides a visual comparison of the performance trends.

The “Relational” Breakthrough. The most striking improvement concerns the *Relational* category. The Zero-Shot model failed almost completely here (F1 0.14), likely because relational stereotypes (e.g., defining a woman solely as a wife) are subtle and structural. The Few-Shot Hybrid strategy, by including specific examples of this category, raised the score to 0.58, proving that In-Context Learning can effectively patch specific semantic blind spots.

Explicit vs. Implicit. All models performed relatively well on *Competence* and *Physical* stereotypes, which are often marked by explicit keywords.

Table 2

Category-wise F1-Scores for the best run of each approach.

Category (Best Run)	Fine-Tuning (Ensemble)	Zero-Shot (V5 - CoT)	Few-Shot (Hybrid k14)
Competence	0.62	0.69	0.67
Personality	0.49	0.56	0.66
Physical	0.58	0.74	0.79
Relational	0.49	0.14	0.58
Role	0.52	0.62	0.65
Sexual	0.32	0.61	0.66

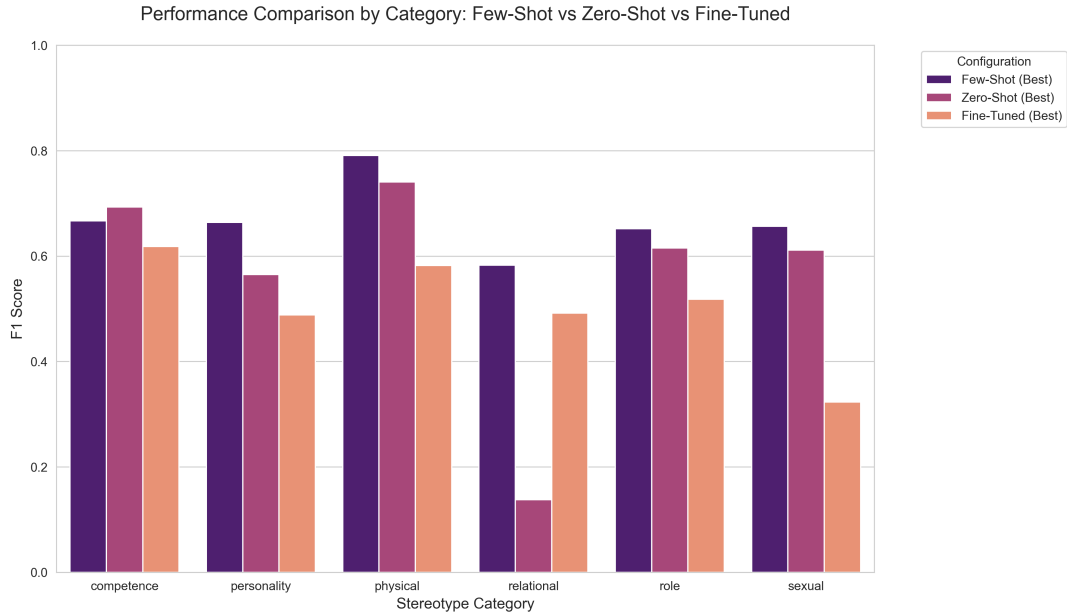


Figure 2: Performance comparison per stereotype category (F1-Score). The **Few-Shot Hybrid** strategy demonstrates superior robustness, particularly in the challenging *Relational* category where Zero-Shot approaches fail. Conversely, Fine-Tuning shows significant degradation in the *Sexual* category due to data scarcity.

Fine-Tuning Failure Cases. The Fine-Tuned model struggled severely with the *Sexual* category (F1 0.32). This suggests that the small training set did not contain enough variety of sexual stereotypes for the model to generalize, whereas the LLM (Gemini) could leverage its vast pre-training knowledge to detect them effectively even without many examples (Zero-Shot F1 0.61).

6. Conclusion

In this paper, we presented a comprehensive study on detecting gender stereotypes in Italian texts for the EVALITA 2026 GSI: detect task. By systematically comparing Fine-Tuning and In-Context Learning approaches, we highlighted the strengths and limitations of current NLP paradigms when applied to low-resource and nuanced tasks.

Our findings indicate that in scenarios with limited training data (200 examples), prompt-based learning with Large Language Models significantly outperforms traditional fine-tuning. Specifically, our **Few-Shot Hybrid Strategy** ($k = 14$) achieved the best performance among the methods we tested (Micro F1: 0.6708), demonstrating that the careful selection of semantic and corrective examples is more impactful than model retraining. Furthermore, our error analysis revealed that while explicit stereotypes (Physical, Competence) are easily detected, implicit biases (Relational) remain a challenge,

requiring targeted interventions like specialized few-shot examples, which successfully bridged the performance gap in our best run.

Declaration on Generative AI

During the development of this work, Generative AI tools were used as support. In particular, some system executions rely on the Google Gemini API (LLM). Additionally, AI-based tools were occasionally employed to help refine code snippets and to improve the clarity of the English writing. All content was carefully reviewed and revised by the author, who takes full responsibility for the correctness and originality of the final report.

Code Availability

The source code, experimental notebooks, and datasets used in this work are publicly available at: <https://github.com/DonatoFe11/gsi-detect-evalita>.

References

- [1] Comandini, G., Speranza, M., Brenna, S., Testa, D., Cavagnoli, S., & Magnini, B. (2026). GSI: detect at EVALITA 2026: Overview of the Task on Detecting Gender Stereotypes in Italian. Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026). Bari, Italy. CEUR.org.
- [2] Cutugno, F., Miaschi, A., Aproso, A. P., Rambelli, G., Siciliani, L., & Stranisci, M. A. (2026). EVALITA 2026: Overview of the 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026). Bari, Italy. CEUR.org.
- [3] Parisi, L., Francia, S., & Magnani, M. (2020). Musixmatch at EVALITA 2020: UmBERTo for Sentiment Analysis. Proceedings of EVALITA 2020.
- [4] Google DeepMind (2024). Gemini: A Family of Highly Capable Multimodal Models. arXiv preprint arXiv:2312.11805.
- [5] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP 2019.

MINDS at GSI:detect: From Logits to Degrees of Agreement in Gender Stereotype Detection with LLMs

Flavio Giobergia

Politecnico di Torino. Turin, Italy

Abstract

We present our submission to the GSI:detect shared task at EVALITA, which focuses on detecting gender stereotypes in Italian texts. Unlike standard binary classification, the task requires predicting a continuous score representing the fraction of annotators who identify a text as containing a gender stereotype. We propose a two-step approach based on few-shot prompting of a large language model and a lightweight regression layer trained on model logits. This design allows us to approximate annotator uncertainty while relying on an open-source model and limited computational resources.

Keywords

annotator disagreement estimation, large language models, few-shot prompting, gender stereotypes

1. Introduction

The detection of gender stereotypes in online contents is an important problem at the intersection of natural language processing (NLP) and social bias analysis. Stereotypes can be expressed explicitly or implicitly, often through subtle linguistic cues, pragmatic assumptions, or culturally grounded generalizations. As a result, their identification is inherently subjective and sensitive to annotator background, values, and lived experience. This makes stereotype detection a particularly challenging task for automated systems, especially when compared to more clearly delimited phenomena such as offensive language or hate speech.

The GSI:detect shared task [1] at EVALITA [2] addresses this challenge for Italian by explicitly embracing subjectivity and disagreement. In this work, we addressed the main task of GSI:detect, and not the subtask of classifying gender stereotypes. Rather than framing stereotype detection as a binary classification problem, the task models it as a regression task, where the target represents the degree of agreement among annotators regarding the presence of gender stereotypes in a text. Each instance is annotated by four annotators, and the gold label – referred to as the *GS value* – corresponds to the fraction of annotators identifying stereotypical content, yielding values in the set $\{0, 0.25, 0.5, 0.75, 1\}$. While constrained by the number of annotators, this formulation is designed to generalize to settings with different annotation schemes and highlights the intrinsically subjective nature of stereotype detection.

This perspective aligns with prior work showing that stereotypes are deeply influenced by subjectivity and social positioning [3], motivating a shift toward perspectivist approaches in NLP [4]. In this view, disagreement among annotators is not noise to be minimized, but rather a signal that reflects how social meaning is constructed and negotiated through language.

Recent years have seen the rapid adoption of large language models (LLMs) across a wide range of NLP tasks, often yielding strong performance even in low-resource or few-shot settings. LLMs have been successfully applied to text classification problems in diverse domains, both in low-resource cases, where limited data is available, but sufficient to fine-tune language models on the smaller end of the scale (e.g., in Italian, for hate speech detection [5], dialect classification [6] [7]), as well as in few-shot scenarios, where only a handful of annotated examples are available: not enough to fine-tune a model,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ flavio.giobergia@polito.it (F. Giobergia)

🆔 0000-0001-8806-7979 (F. Giobergia)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

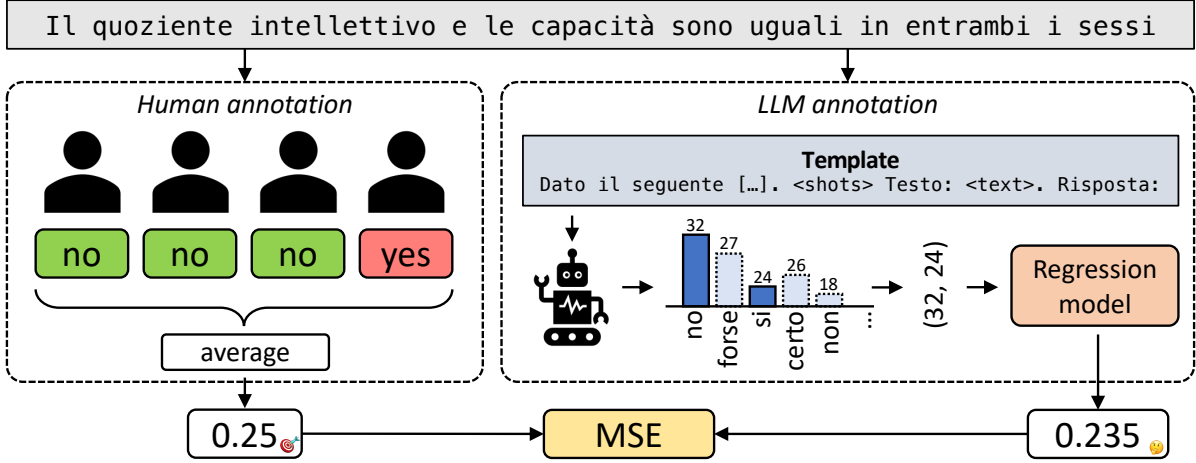


Figure 1: Summary of the human and LLM-based annotation of the target data. The input sentence (in English, “The intelligence quotient and the capabilities are the same for both genders”) is annotated by four humans as either containing gender stereotypes (yes) or not (no). The GS value is computed as the average score (no = 0, yes = 1). The LLM is prompted to produce a similar annotation. The logits produced by the LLM are mapped, with a regression model, to the target GS value (using the limited training set available).

but sufficient to use larger models with an In-Context Learning approach [8]. This is particularly useful, for instance, when annotations are expensive or can only be carried out by specialized domain experts, as is the case for the verification of scientific claims [9], or the annotation of scientific works [10].

These applications suggest that LLMs capture rich semantic and pragmatic information, making them promising candidates for socially grounded tasks that require sensitivity to nuance and ambiguity. At the same time, most prior work focuses on categorical decisions, whereas fewer studies investigate whether LLMs can meaningfully model degrees of uncertainty or disagreement inherent in subjective annotations.

In this work, we propose a simple yet effective approach that leverages a large language model in a few-shot setting and exploits its internal confidence signals, as represented by predicted logits, to approximate annotator agreement. Rather than relying solely on discrete predictions, we treat these signals as proxies for uncertainty and calibrate them through a regression model. This allows us to capture both the presence of gender stereotypes and the noisy nature of annotator consensus.

2. Methodology

We summarize the annotation pipeline in Figure 1. The annotators’ scores are summarized into their average. The LLM annotation is produced using a template that embeds a limited number of randomly sampled shots, and the text to be annotated. The output of the LLM is processed by (1) extracting the relevant logits and (2) building a regression model that maps logits to the final GS value. The rest of the section provides further details on the proposed approach.

2.1. Few-shot prompting

We employ a large language model in a few-shot learning setup. The model is prompted with a short description of the task, as well as 20 randomly selected examples illustrating gender-stereotypical and non-stereotypical content.

Importantly, the model is *not* asked to directly predict the numerical target value: it is well known that LLMs have limitations when it comes to handling numerical values, due to their operations at a token-level [11].

Instead, the model is prompted directly to produce a yes/no¹ outcome regarding the presence of

¹Since the model is prompted in Italian, the actual question requires an annotation as *si* (yes) or *no*. The rest of the paper uses

stereotypical contents in the input text, and the 20 random shots provided in the prompt are binarized through a thresholding ('yes' if the probability is greater than or equal to 0.5, 'no' otherwise).

Upon generation of an answer, instead of directly using the generated answer, we extract the logits associated with the possible responses (namely, 'si' and 'no', since the model is prompted in Italian). We assume that the magnitudes associated with these two logits are representative of the model's internal confidence – which, we argue, is proportional to the agreement among annotators. In other words, for ambiguous text (i.e., GS value ≈ 0.5) the LLM is expected to assign similar probabilities to the 'yes' and 'no' tokens, whereas for a text for which annotators are more confident (i.e., GS value ≈ 0 or GS value ≈ 1), the LLM is expected to assign higher probability to the correct token.

2.2. Logit-based regression

Rather than relying on the model's direct output as the final prediction, we introduce a calibration step. The extracted logits for the 'yes' and 'no' outcomes are treated as separate features, used to train a lightweight regression model (e.g., linear regression or K-Nearest Neighbors) on the training set. The regression target is the fractional agreement score derived from the human annotations (i.e., the GS value).

This two-step process allows us to map the model's logits to a continuous prediction that reflects annotator consensus. Intuitively, larger differences between logits indicate higher certainty, whereas smaller differences correspond to ambiguous cases – mirroring the variability observed among human annotators.

2.3. Inference

At inference time, a new text is first processed by the language model under the same few-shot prompt. The resulting logits are then passed through the trained regression model to produce a final predicted score in $[0, 1]$. This score represents the estimated fraction of annotators who would identify the text as containing a gender stereotype.

3. Experimental Results

In this section, we report the experimental results obtained with the proposed system. We selected Qwen 2.5 14B [12] as a trade-off between representational capacity and computational constraints (single-GPU inference, no fine-tuning), observing consistent gains over smaller open models. As such, we report all results with the specified Qwen model.

The dataset available included 200 annotated samples, and 810 test samples used for the final evaluation. As already discussed, we used 10% of the development set (i.e., 20 randomly chosen samples) for the generation of the prompt, and we adopted the remaining 180 samples for the training of the regression model that predicts the final agreement, based on the yes/no logits.

Table 1 reports the official results of the GSI: detect shared task, in terms of Mean Squared Error (MSE), Normalized MSE (NMSE) and the inverse of $1 + \text{NMSE}$ (for a *higher-is-better* metric). Our system ranks approximately in the middle of the leaderboard. While it does not outperform all other approaches, it achieves competitive performance when compared against similar baselines, such as the baseline model using Qwen 3 (14B), and even outperforms, in some cases, commercial baselines (e.g., based on GPT-5 nano).

Prompt details. We note that the runs that were submitted for the competition contained a misleading prompt, focused on misogyny rather than gender stereotype detection (entries marked as (*mys*) in Table 1). This experiment was carried out as a part of the prompt engineering process. We additionally report the (non official) results, obtained with the revised prompt (marked as (*GS*) in Table 1). This small, but

yes/no or si/no interchangeably.

Table 1

Results overview with ranking, models, and evaluation metrics. Given the large number of participants, only the top-3 solutions submitted by other participants (DIAG-Sapienza, StereoBusters) are reported. An asterisk is used to highlight non-official solutions (i.e., solutions that have *not* been submitted for the competition). (*GS*) marks solutions with a gender stereotype-focused prompt, whereas (*mys*) marks solutions with a misogyny-focused one.

Rank	Team	Model	Approach	MSE ↓	NMSE ↓	$\frac{1}{1+NMSE}$ ↑
1	DIAG-Sapienza	GPT-5	zero-shot	0.0766	0.4292	0.700
2	DIAG-Sapienza	GPT-5	few-shot (4)	0.0820	0.4594	0.685
3	StereoBusters	Gemma 3 (27B)	few-shot (5)	0.0983	0.5508	0.645
-	* <i>MINDS (GS)</i>	Qwen 2.5 (14B) + KNN	few-shot (20)	0.1104	0.6184	0.618
16	BASELINE	GPT-5 nano (split prompt)	zero-shot	0.1150	0.6430	0.609
-	* <i>MINDS (GS)</i>	Qwen 2.5 (14B) + LR	few-shot (20)	0.1192	0.6676	0.600
19	BASELINE	GPT-5 nano (unified prompt)	zero-shot	0.1240	0.6950	0.590
22	<i>MINDS (mys)</i>	Qwen 2.5 (14B) + KNN	few-shot (20)	0.1288	0.7213	0.581
-	* <i>MINDS (GS)</i>	Llama 3 (8B) + LR	few-shot (20)	0.1329	0.7446	0.573
-	* <i>MINDS (GS)</i>	Llama 3 (8B) + KNN	few-shot (20)	0.1363	0.7636	0.567
27	<i>MINDS (mys)</i>	Qwen 2.5 (14B) + LR	few-shot (20)	0.1379	0.7725	0.564
37	BASELINE	Qwen 3 (14B)	zero-shot	0.1500	0.8420	0.543
40	BASELINE	prediction=0.5	-	0.1800	1.0083	0.498

important prompt detail shows a relevant change in performance (for the best-performing model, a 14% improvement in terms of MSE, from 0.1288 down to 0.1104).

Additional models. We include the performance obtained with a smaller model (Llama 3 8B [13]) to show that the larger Qwen model consistently achieves better performance w.r.t. smaller versions (when considering the “GS” solutions): a potentially larger model, therefore, could be expected to achieve even higher performance.

Regression approach. As for the regression model, we include two alternatives: a simple linear regression (LR), and KNN (with $K = 15$, selected following a hyperparameter tuning step). We show that both regression models produce similar results, with slight variations. KNN provides the best performance for Qwen, although this is not necessarily always the case, as confirmed in Section 3.2.

3.1. Predictive capabilities of logits

We provide a qualitative result that supports the choice of using the yes/no logits for the prediction of the agreement. In particular, we report in Figure 2 the scatter plot of the yes and no logits for the 180 samples with known ground truth, as well as for the 810 unlabelled samples. The labelled samples are colored according to the ground truth agreement: it can be observed that high-confidence results (i.e., large logit values for one class and small logit for the other one) are associated with higher agreement (either close to 0, or close to 1 depending on the outcome).

3.2. Number of shots

One aspect of interest for the choice of In-Context Learning approaches is the quantity of shots to be used for the prompt. In general, a large number of shots (e.g., as many as can fit the context window) can be used. However, for this specific system, there is a trade-off to be made. The limited number of labelled points available is used both for the preparation of the prompt, as well as for the calibration through the training of the regression model. So, using an excessive number of examples in the prompt would not leave sufficient samples for the second step. We measure the performance of the solution for a varying number of shots provided in the context (from 0 to 32). Figure 3 shows the evolution, using as heads both a linear regression, and a KNN regressor, over 5 runs with different randomly sampled shots.

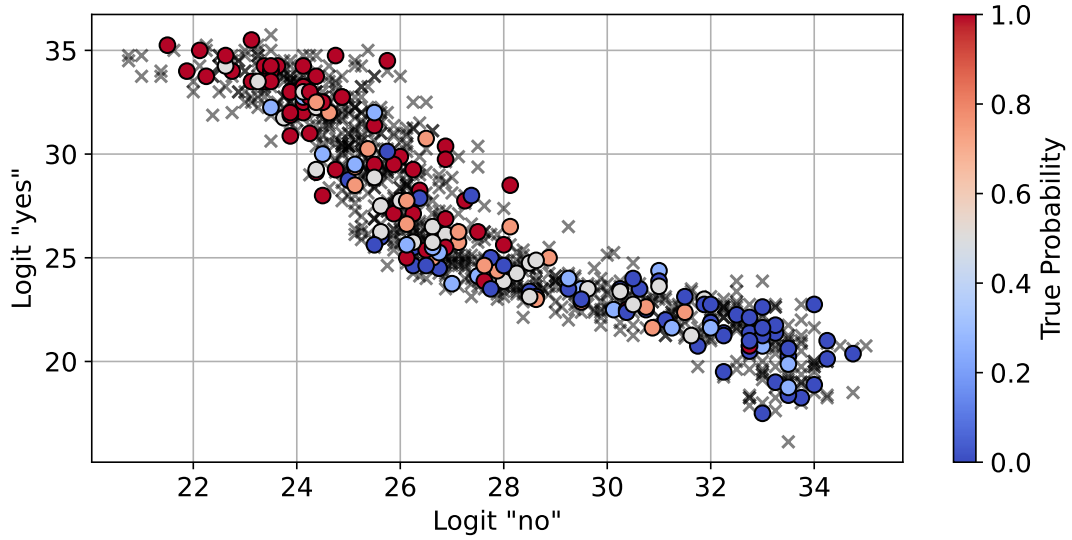


Figure 2: Scatter plot of the no/yes logits produced by the LLM when asked to annotate each sample. The color is proportional to the ground truth value for the “train” samples, whereas the ‘x’ marker represents “test” samples.

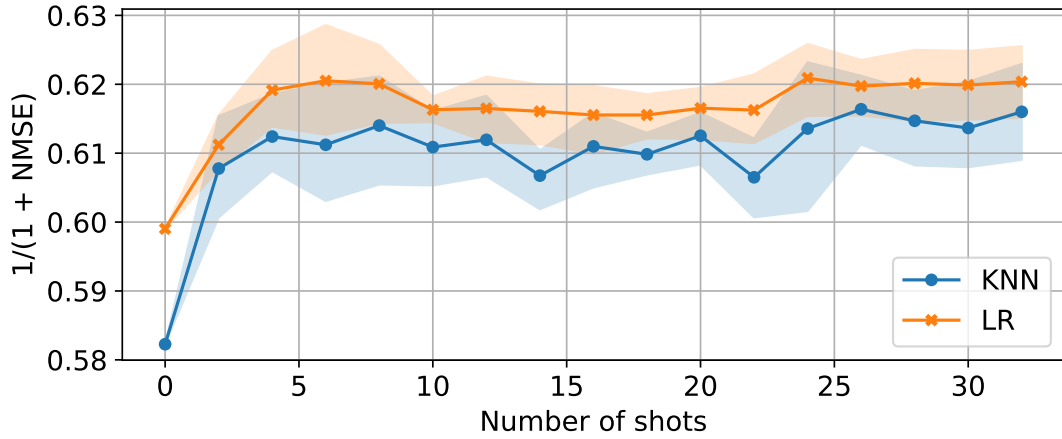


Figure 3: Change in performance, measured as $1/(1+NMSE)$, as a function of the number of shots provided. The error bars represents the 95% confidence intervals over 5 runs, with randomly samples shots.

Two interesting aspects emerge. First, the 95% confidence interval overlap between KNN and LR, although the mean performance observed for LR w.r.t. KNN suggests consistent superiority of LR, without strong statistical separation. Second, the measured performance improves when passing a non-zero number of shots, but it does not improve significantly when using more than 4-8 shots. The choice of using 20 shots for the submissions, therefore, may not have been ideal. However, these are considerations that can only be made in hindsight: during the competition, the limited labelled dataset available was used to run a similar experiment, producing different results – which seemed to indicate that a larger number of shots could be beneficial. However, the limited validation set size used and the execution of a single run may have produced less reliable results.

4. Conclusions

We presented a two-step approach for the GSI:detect shared task that combines few-shot prompting of a large language model with a regression-based calibration layer. By leveraging model logits as a proxy for confidence, our method captures the graded nature of annotator agreement and provides a nuanced

prediction of gender stereotype presence.

Our experiments show that this approach outperforms direct score prediction and achieves competitive results using an open-source model at a fraction of the cost of large proprietary systems. Future work could explore more advanced calibration techniques, alternative uncertainty-aware objectives, and extensions to other bias-related tasks and languages.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (GPT 5.2) in order to: Drafting content. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini, Gsi:detect at evalita 2026: Overview of the task on detecting gender stereotypes in italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, M. A. Stranisci, An italian twitter corpus of hate speech against immigrants, in: Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018), 2018.
- [4] F. Cabitza, A. Campagner, V. Basile, Toward a perspectivist turn in ground truthing for predictive computing, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, 2023, pp. 6860–6868.
- [5] M. Sanguinetti, G. Comandini, E. Di Nuovo, S. Frenda, M. Stranisci, C. Bosco, T. Caselli, V. Patti, I. Russo, Haspeede 2@ evalita2020: Overview of the evalita 2020 hate speech detection task, Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (2020).
- [6] A. Ramponi, C. Casula, Geolingit at evalita 2023: Overview of the geolocation of linguistic variation in italy task, in: EVALITA Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian Final Workshop: Parma, Italy, September 7-8th, 2023, Accademia University Press, 2024, p. 109.
- [7] A. Koudounas, F. Giobergia, I. Benedetto, S. Monaco, L. Cagliero, D. Apiletti, E. Baralis, et al., ba ρ tti at geolingit: Beyond boundaries, enhancing geolocation prediction and dialect classification on social media in italy, in: CEUR workshop proceedings, CEUR, 2023.
- [8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

- [9] C. Alvarez, M. Bennett, L. L. Wang, Zero-shot scientific claim verification using llms and citation text, in: *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, 2024, pp. 269–276.
- [10] F. Giobergia, A. Koudounas, E. Baralis, Large language models-aided literature reviews: A study on few-shot relevance classification, in: *2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT)*, IEEE, 2024, pp. 1–5.
- [11] H. Li, X. Chen, Z. Xu, D. Li, N. Hu, F. Teng, Y. Li, L. Qiu, C. J. Zhang, L. Qing, et al., Exposing numeracy gaps: A benchmark to evaluate fundamental numerical abilities in large language models, in: *Findings of the Association for Computational Linguistics: ACL 2025*, 2025, pp. 20004–20026.
- [12] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, *arXiv preprint arXiv:2505.09388* (2025).
- [13] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., The llama 3 herd of models, *arXiv preprint arXiv:2407.21783* (2024).

StereoBusters at GSI:detect: LLM-Based Detection and Human Qualitative Analysis of Gender Stereotypes in Italian Short Texts

Salvatore Greco¹, Moreno La Quatra², Marta Marchiori Manerba³, Ricardo Muñoz Sánchez⁴ and Alessandra Teresa Cignarella⁵

¹Centre for Data Futures, King's College London, United Kingdom

²Speech Technologies and Machine Learning Lab, Kore University of Enna, Italy

³Computer Science Department, University of Turin, Italy

⁴Språkbanken Text, University of Gothenburg, Sweden

⁵Language and Translation Technology Team (LT3), Ghent University, Belgium

Abstract

This paper presents the participation of the StereoBusters team in GSI:detect, a novel shared task at EVALITA 2026 focused on detecting and classifying gender stereotypes in short Italian texts. The competition consists of a main regression task that estimates the degree of gender-stereotypical content in a text, and a subtask that performs fine-grained classification into six gender stereotype categories. We describe our submitted systems in two settings: zero-shot and few-shot. We propose two LLM-based approaches: (1) using a single LLM to directly predict the degree of gender-stereotypical content and its category, and (2) using a panel of four LLMs, each predicting a binary gender-stereotypical label, and aggregating these predictions to obtain a continuous degree of stereotypical content, while using the best model in the panel to predict the category. We evaluate five prompting strategies: direct, guideline-based, Chain-of-Thought, Italian-language, and annotator-perspective prompting.

Our results show that: (1) while the approach aggregating a panel of LLMs performs better on the development data than a single continuous LLM approach, it generalizes less effectively on the test data; (2) prompts tend to benefit from the inclusion of annotation guidelines; and (3) even open-weight LLMs with fewer parameters can outperform larger models across both tasks and proposed approaches. In the official leaderboard, our best system ranked 2nd in the zero-shot setting of the main task ($nMSE = 0.626$), and 2nd in the few-shot setting of the main task ($nMSE = 0.645$). In the subtask, our best system ranked 1st in the zero-shot setting ($F1 \text{ micro} = 0.646$), and 3rd in the few-shot setting ($F1 \text{ micro} = 0.669$). Additionally, following the annotation guidelines, we qualitatively analyse the 200 development texts and provide additional annotations.

Keywords

Stereotype Detection, Gender Stereotypes, Italian, Prompting, Guidelines, LLMs, NLP

Trigger warning: This paper contains examples of stereotypical and potentially triggering content.

1. Introduction

The exponential growth of social media has transformed online communication, enabling users to share opinions on virtually any topic at any time. However, this surge has also amplified the spread of harmful content, including hateful comments and prejudiced discourse. Such phenomena often stem from deeply ingrained stereotypes, which perpetuate discrimination and reinforce structural inequalities.

Natural Language Processing (NLP) can play a crucial role in addressing these challenges by providing tools to analyse linguistic manifestations of bias and automate their detection at scale. Within the Italian NLP community, several initiatives such as HaSpeeDe [1, 2], AMI [3, 4], HODI [5], and CALAMITA [6] have focused on hate speech detection and related phenomena, laying the groundwork for more nuanced investigations. Some works specifically focused on racial and ethnic stereotypes in Italian [7, 8, 9, 10]. Finally, other studies have explored NLP approaches for promoting gender-inclusive language, focusing

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
✉ salvatore.greco@kcl.ac.uk (S. Greco); moreno.laquatra@unikore.it (M. La Quatra); marta.marchiorimanerba@unito.it (M. Marchiori Manerba); ricardo.munoz.sanchez@gu.se (R. Muñoz Sánchez); alessandrateresa.cignarella@ugent.be (A. T. Cignarella)

ORCID 0000-0001-7239-9602 (S. Greco); 0000-0001-8838-064X (M. La Quatra); 0000-0003-2251-1824 (M. Marchiori Manerba); 0000-0002-9902-2925 (R. Muñoz Sánchez); 0000-0002-4409-6679 (A. T. Cignarella)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

on rewriting or translating texts in a more gender-neutral manner [11, 12, 13, 14, 15, 16]. These efforts contribute to mitigating harmful content and fostering safer, more inclusive online spaces.

Despite these advances, research specifically targeting gender stereotypes in Italian texts remains scarce, compared to prior work on English and other high-resource languages. Furthermore, existing approaches often assume a binary conception of gender and lack robust evaluation baselines [17, 18]. The GSI:DETECT shared task [19], introduced at EVALITA 2026 [20], represents the first initiative in this domain for Italian. Its goal is to promote research on identifying stereotypical representations of gender across diverse short-text genres. To this end, the organizers released a manually curated dataset comprising 1,010 texts from social media and websites, annotated by four expert annotators to capture both aggregated and individual perspectives. The task is articulated into two parts:

- **Main task:** a regression task to determine the degree of gender-stereotypical content in a text.
- **Subtask:** a fine-grained classification task requiring systems to predict one of six stereotype categories.

In this paper, we present the systems developed by the **StereoBusters team** for the two tasks of GSI:DETECT. We explore the first two of the three experimental settings: *zero-shot*, *few-shot* and *fine-tuning*. We propose and evaluate two approaches leveraging large language models (LLMs) and diverse prompting strategies, including guideline integration and simulated annotator panels. We also performed an additional annotation experiment and a qualitative analysis (discussed in Appendix C).¹

2. Task Description and Dataset

Task Description The GSI:DETECT task comprises (1) a main task to quantify the degree of Gender Stereotype (*GS value*) in a text; and (2) a subtask to classify the Gender Stereotype category (*GS category*).

Detecting stereotypes, however, is an inherently complex and subjective task, and the perception of stereotypes varies significantly across individuals [21]. For this reason, the GSI:DETECT task follows the perspectivist approach [22, 23]. Rather than relying on a single label obtained through majority voting, the dataset preserves non-aggregated annotations and provides a score designed to capture this variation. As a result, the *GS value* is defined as a single numerical value between 0 and 1 that reflects the fraction of annotators who labeled a text as containing stereotypes.

The *GS category* identifies the stereotype type in each text. Annotators assigned one category per text; if multiple applied, they chose the first occurring one. Categories are the following: (1) **Role stereotypes** encompass sociocultural expectations of the roles that people are expected to play regarding family or society; (2) **Personality stereotypes** attribute emotional, behavioral, or character traits to people based on their (perceived) gender; (3) **Competence stereotypes** are generalized judgments about professional, intellectual, or physical skills and abilities regardless of gender; (4) **Physical stereotypes** reflect expectations about the physical aspect of someone based on their (perceived) gender; (5) **Sexual stereotypes** are related to desire, sexuality, and sexual behavior. Insults with sexual connotations are also included in this category, regardless of whether they are direct or indirect; (6) **Relational stereotypes** refer to expectations of how people are expected to connect in interpersonal relationships, depending on their gender, including affective, romantic, and/or sexual relationships; (7) **No stereotype** means that none of the annotators considered that the text contains stereotypes.

Overview of the GSI:detect Dataset The dataset contains 1,010 short Italian texts: 200 in the development set and 810 in the test set. Texts were collected from social media and information websites to cover both informal and formal registers (a more detailed data analysis is discussed in Appendix A).

A total of four annotators sensitive to gender-related issues were involved in the annotation process. Each annotator labeled all texts by assigning (1) a binary yes/no gender stereotype label and (2) at most one stereotype category. Annotators were instructed to select the appropriate stereotype category when a text was judged to contain stereotypes, or to choose the “no” category when no stereotypes were present. If multiple stereotype categories appeared within the same text, annotators selected the category that occurred first in the text. The final GS value is computed by converting the four binary gender stereotype labels into numerical values (1 for “yes” 0 for “no”) and taking their mean. The final GS category is determined via majority voting among annotators’ category labels.

¹The code repository is available at <https://github.com/grecosalvatore/StereoBusters-GSI-Detect-Evalita2026>.

3. Methodology

3.1. Gender Stereotypes Detection Approaches

Given the relatively limited size of the development data, our approach relies on zero-shot evaluation and in-context learning, rather than direct parameter tuning. However, since the dataset is annotated not only with an aggregate value but also with binary stereotype labels provided by individual annotators, we propose two alternative prediction targets for the *GS value* (main task):

Approach 1 (CONTINUOUS): A single LLM is prompted to directly predict the aggregated GS value—the continuous gender stereotype score in $[0, 1]$. The LLM is expected to capture the degree of stereotypicality by implicitly modeling potential disagreement and subjectivity across multiple diverse viewpoints.

Approach 2 (BINARY): Multiple LLMs are prompted to predict a binary gender stereotype label—whether the text contains a gender stereotype (0 = no, 1 = yes). The GS value is obtained by aggregating the binary predictions of multiple models. Specifically, we construct a panel of four models to mimic this process. Each model in the panel corresponds to a specific model-prompt combination selected to best replicate the labeling behavior of an individual annotator.

In both approaches, the *GS category* is predicted jointly with the main target when a stereotype is present. Under the CONTINUOUS approach, a single LLM is instructed to directly predict the GS value and, when the GS value is ≥ 0.25 , to also assign a stereotype category. In contrast, under the BINARY approach, each LLM predicts a binary gender-stereotype label and, if a stereotype is detected, additionally assigns its category. For the BINARY setting, the final GS value is obtained by aggregating the four binary predictions from the LLM panel, while the final category is determined either via majority vote or by designating one LLM as an oracle (i.e., super judge).

3.2. Prompting Strategies

For both approaches, we design and evaluate the following prompting strategies, reported in Figure 2 in Appendix B for the CONTINUOUS approach and in Figure 3 in Appendix B for the BINARY approach:

1. **Direct Prompt:** The LLM is directly instructed to assess whether the input text contains gender stereotypes and to output the corresponding prediction, with minimal additional guidance. Depending on the approach, the output is either a continuous GS value (Figure 2a) or a binary gender-stereotype label (Figure 3a).
2. **Chain-of-Thought (CoT) Prompt:** The LLM is instructed to reason step by step using a Chain-of-Thought strategy [24] before producing the final prediction (Figure 2b and Figure 3b). The underlying idea is that explicitly modeling intermediate reasoning steps may help the model better identify implicit gender stereotypes and arrive at more accurate predictions.
3. **Guidelines-Informed Prompt:** The LLM is provided with additional task-specific information, including detailed category definitions and short illustrative examples drawn directly from the annotation guidelines (Figure 2c and Figure 3c). The underlying idea is that explicitly incorporating the original annotation criteria may better align the model’s predictions with human judgments.
4. **Italian Prompt:** The LLM is instructed to assess whether the input text contains gender stereotypes and to output the corresponding prediction, with minimal additional guidance. However, in this case, the prompt is written in Italian. The underlying idea is that, since the task involves detecting gender stereotypes in Italian texts, using the target language may better align the model’s reasoning with linguistic and cultural cues specific to Italian (Figure 2d and Figure 3d).
5. **Perspectivist Prompt:** The LLM is instructed to emulate a panel of four annotators with differing sensitivities and perspectives. One annotator is highly sensitive to subtle stereotypes, two are moderately sensitive, and one identifies only explicit stereotypes. Note that this prompt is implemented only in the continuous setting (Figure 2e), as this approach mirrors the aggregation of the four binary models used to construct the panel for the binary approach.

All prompts predict the GS value or the binary gender stereotype label together with the GS category. Models are instructed to assign the “no” category when the GS value or binary stereotype label is 0.

3.3. Zero-shot and Few-shot Settings

All prompts are evaluated in both *zero-shot* and *few-shot* learning. The prompt structure is identical across the two settings; however, in *few-shot*, five illustrative examples are introduced in each prompt to guide the model’s prediction, as shown in the red text in Figure 2f in Appendix B.

Given an input text, the five examples for the *few-shot* setting are identified by following these steps: (1) We split the annotated samples in the development set into two subsets: samples *with context* and samples *without context*; (2) We compute semantic embedding of the input text using the PARAPHRASE-MULTILINGUAL-MPNET-BASE-V2 model; (3) We retrieve the 5 most similar examples from the same subset (with or without context) based on cosine similarity between embeddings; and (4) The selected examples are formatted with their *text*, *GS value* for the CONTINUOUS approach or *binary gender stereotype label* for the BINARY approach (specific to one annotator), and *category*, and included in the system prompt.

The main differences between the *few-shot* implementations of the CONTINUOUS and BINARY approaches are as follows: (1) the ground-truth label provided in the examples differs, as the continuous version predicts the GS value, whereas the binary version predicts a binary gender stereotype label; and (2) the continuous approach needs to be executed only once per input text in the *few-shot* setting, since it produces a single aggregated score, while the binary approach must be run separately to emulate each specific annotator, as each sample has a binary stereotype label assigned by each annotator.

4. Results on the Development Data

In this section, we present the experimental results obtained on the development data. We first discuss the preliminary experiments conducted to identify the panels for the BINARY approach (Section 4.1), followed by the results for GS value prediction (Section 4.2) and GS category classification (Section 4.3).

Models We evaluated non-commercial, open-weight LLMs from different families and sizes: Llama 3 [25, 26] (8B, 70B), Qwen [27] (8B, 32B), LLaMAntino [28, 29] (8B), and Gemma 3 [30] (4B, 12B, 27B).

4.1. Preliminary Experiments for Panels Compositions

This experiment aims to identify the most effective combinations of models and prompts to mimic each annotator in the binary gender stereotype labeling task, following the proposed BINARY approach.

Evaluation metrics We compute macro-averaged F1, Precision, and Recall for the prediction of the binary gender stereotype label, evaluated separately for each annotator using their corresponding binary ground-truth labels (0/1 for absence/presence of stereotype). In addition, we assess the agreement between the LLM predictions and each annotator by computing the Cohen’s Kappa (κ) coefficient [31].

Results Table 1 reports the macro F1, Precision (P), Recall (R), and Cohen’s κ scores for the binary gender stereotype prediction task. Results are presented separately for each annotator (A1–A4). In the *zero-shot* setting, all model–prompt combinations are evaluated. In contrast, in *few-shot*, only the best-performing model–prompt combination for each model within the selected panels is tested.²

Zero-shot Results In the *zero-shot* setting (top section of Table 1), all model–prompt combinations achieve high F1 scores and annotator agreement, except for the CoT prompt, which consistently yields lower performance. The best model–prompt combinations based on Cohen’s κ are LLAMA-3.3-70B-INSTRUCT with the Italian prompt for A1, LLAMA-3.1-8B-INSTRUCT with the Direct prompt for A2, LLAMANTINO-3-ANITA-8B with the Guidelines prompt for A3, and GEMMA-3-27B-IT with the Direct prompt for A4. When optimizing for F1, the best combinations are LLAMA-3.3-70B-INSTRUCT with the Direct prompt for A1, LLAMA-3.3-70B-INSTRUCT with the Italian prompt for A2, GEMMA-3-12B-IT with the Direct prompt for A3, and GEMMA-3-27B-IT with the Direct prompt for A4. Interestingly, even smaller models with 8B or 12B parameters showed higher agreement with the annotators.

²As discussed above, in the binary approach, the ground-truth labels are annotator-specific. Consequently, each model–prompt combination must be evaluated separately for each annotator in *few-shot*, resulting in a very large experimental space.

Table 1

Results on the Development Data for Binary Gender Stereotype Detection (BINARY Approach). Performance comparison of models in modeling individual annotators, used to identify the panels for the BINARY approach. Results are reported per annotator (A1–A4) and include F1 score, Precision (P), Recall (R), and Cohen’s Kappa (κ), evaluated separately under *zero-shot* and *few-shot* settings. In *few-shot*, only the best-performing models from the *zero-shot* experiments were evaluated. Best-performing *zero-shot* results for each metric are in bold and underscored. Model-prompt combinations selected for **Panel 1** (maximizing Cohen’s κ) and **Panel 2** (maximizing F1) are indicated with colored labels [P1Ax] and [P2Ax], where x denotes the annotator being modeled, with corresponding scores highlighted in gray.

		Per-Annotator Prediction															
Model	Prompt	A1				A2				A3				A4			
		F1	P	R	κ	F1	P	R	κ	F1	P	R	κ	F1	P	R	κ
Zero-Shot																	
Llama-3.1-8B-Instruct	Direct [P1A2]	.73	.74	.74	.47	.74	.76	.76	.50	.75	.76	.76	.49	.76	.75	.76	.51
	CoT	.55	.71	.61	.21	.49	.67	.59	.15	.60	.74	.63	.27	.62	.76	.64	.31
	Guidelines	.69	.72	.70	.40	.70	.75	.74	.44	.73	.74	.72	.46	.76	.77	.75	.52
	Italian	.69	.70	.70	.40	.69	.72	.72	.41	.74	.75	.74	.49	.74	.75	.74	.48
Llama-3.3-70B-Instruct	Direct [P2A1]	.75	.75	.75	.50	.74	.75	.76	.49	.72	.72	.72	.44	.74	.74	.74	.48
	CoT	.72	.77	.73	.46	.66	.73	.71	.37	.72	.75	.72	.45	.72	.75	.72	.45
	Guidelines	.70	.74	.72	.43	.69	.74	.72	.41	.72	.74	.72	.45	.75	.77	.75	.51
	Italian [P1A1] - [P2A2]	.75	.75	.75	.51	.75	.75	.76	.50	.70	.71	.71	.41	.73	.74	.74	.47
Gemma-3-4B-it	Direct	.65	.69	.66	.32	.62	.68	.66	.30	.67	.69	.67	.35	.69	.71	.69	.39
	CoT	.34	.57	.50	.01	.31	.54	.50	.00	.38	.62	.51	.02	.37	.45	.50	-.01
	Guidelines	.67	.69	.68	.36	.65	.69	.68	.34	.69	.70	.69	.38	.73	.74	.73	.46
	Italian	.58	.72	.63	.26	.55	.72	.63	.23	.65	.78	.67	.36	.64	.75	.65	.33
Gemma-3-12B-it	Direct [P2A3]	.72	.73	.73	.45	.73	.74	.74	.46	.76	.76	.76	.51	.74	.73	.74	.47
	CoT	.67	.72	.69	.37	.62	.68	.66	.29	.66	.69	.66	.34	.67	.69	.67	.35
	Guidelines	.70	.73	.71	.42	.68	.74	.72	.40	.75	.77	.74	.50	.77	.78	.76	.54
	Italian	.71	.74	.72	.44	.68	.73	.72	.40	.74	.75	.74	.48	.76	.77	.75	.52
Gemma-3-27B-it	Direct [P1A4] - [P2A4]	.69	.73	.71	.41	.68	.73	.71	.39	.73	.75	.73	.47	.78	.80	.78	.57
	CoT	.63	.71	.66	.32	.60	.69	.65	.27	.70	.76	.70	.42	.70	.75	.70	.42
	Guidelines	.67	.72	.69	.37	.66	.73	.70	.36	.72	.75	.71	.44	.77	.80	.76	.54
	Italian	.67	.71	.68	.36	.65	.71	.69	.35	.73	.76	.73	.48	.77	.79	.76	.54
Qwen3-8B	Direct	.65	.71	.67	.34	.69	.73	.69	.40	.62	.70	.66	.30	.62	.71	.67	.31
	CoT	.69	.70	.69	.38	.69	.69	.69	.38	.62	.65	.64	.27	.61	.64	.63	.25
	Guidelines	.51	.57	.54	.09	.58	.64	.59	.21	.54	.65	.60	.18	.53	.64	.60	.17
	Italian	.67	.70	.67	.35	.72	.74	.71	.44	.64	.69	.67	.32	.66	.72	.70	.36
Qwen3-32B	Direct	.71	.72	.71	.43	.74	.74	.74	.47	.73	.75	.75	.47	.72	.74	.74	.45
	CoT	.72	.74	.73	.46	.69	.73	.72	.41	.70	.71	.70	.40	.74	.75	.74	.48
	Guidelines	.71	.72	.71	.43	.73	.73	.73	.46	.70	.71	.71	.41	.72	.74	.74	.45
	Italian	.69	.70	.69	.38	.71	.72	.71	.42	.70	.73	.72	.42	.67	.70	.69	.36
LLaMAntino-3-ANITA-8B	Direct	.71	.71	.71	.42	.71	.72	.72	.43	.75	.75	.76	.51	.74	.74	.75	.49
	CoT	.68	.69	.68	.37	.68	.71	.71	.38	.70	.70	.69	.39	.73	.73	.72	.45
	Guidelines [P1A3]	.71	.71	.71	.42	.71	.72	.73	.43	.76	.76	.76	.52	.75	.75	.75	.50
	Italian	.68	.70	.69	.38	.67	.71	.70	.38	.74	.75	.74	.48	.77	.78	.77	.54
Few-Shot																	
Llama-3.1-8B-Instruct	Direct [P1A2]	-	-	-	-	.61	.73	.68	.31	-	-	-	-	-	-	-	-
Llama-3.3-70B-Instruct	Direct [P2A1]	.73	.75	.74	.47	-	-	-	-	-	-	-	-	-	-	-	-
Llama-3.3-70B-Instruct	Italian [P1A1] - [P2A2]	.74	.74	.74	.49	.77	.78	.77	.54	-	-	-	-	-	-	-	-
Gemma-3-12B-it	Direct [P2A3]	-	-	-	-	-	-	-	-	.76	.76	.75	.51	-	-	-	-
Gemma-3-27B-it	Direct [P1A4] - [P2A4]	-	-	-	-	-	-	-	-	-	-	-	-	.73	.74	.73	.47
LLaMAntino-3-ANITA-8B	Guidelines [P1A3]	-	-	-	-	-	-	-	-	.73	.75	.73	.47	-	-	-	-

Few-shot Results In the *few-shot* setting (bottom section of Table 1), surprisingly, most models exhibit a decrease in the F1 score and lower Cohen’s κ agreement with the corresponding annotators. The only exception is LLaMA-3.3-70B-INSTRUCT (Italian), which shows improved performance for Annotator 2. The decrease in performance in the few-shot setting may be attributed to the binary nature of the gender stereotype detection task in the BINARY approach, where few examples can overly bias the model’s decision boundary, particularly for smaller models.

Summary of Results The four best-performing model–prompt combinations to mimic individual annotators in the development data, were grouped into two panels (Table 2b). These panels were then used to compute the GS values in our development experiments and for the final submissions.

4.2. GS Value Prediction Results on Development Data (Main task)

This experiment aims to evaluate *GS value* prediction (main task) on development data across all models using the CONTINUOUS approach, and the two panels identified in Section 4.1 for the BINARY approach.

Evaluation metrics We evaluate model performance using the complements of the Mean Squared Error (MSE) and the Mean Absolute Error (MAE), which quantify the discrepancy between the predicted and ground-truth GS values. Since GS values are continuous and lie in the interval $[0, 1]$, we report the metrics as $(1 - \text{MSE})$ and $(1 - \text{MAE})$, with higher values indicating better predictive performance.

Results Table 2a reports the complements of the MSE and MAE in *zero-shot* and *few-shot* settings for

all model-prompt combinations for the CONTINUOUS approach and for the two BINARY panels.

Zero-shot Results In the *zero-shot* setting, all model-prompt combinations perform well (1 - MSE ranging from 0.727 to 0.898). The best-performing systems are, for the CONTINUOUS approach: (1) LLAMA-3.3-70B-INSTRUCT with guidelines-informed prompt and (2) GEMMA-3-12B-IT with Italian prompt; and, for the BINARY approach, (3) Panel 1 based on Cohen’s κ . All three systems achieve the highest score of 0.898. Other model-prompt combinations in the CONTINUOUS setting achieve very similar performance, such as LLAMA-3.3-70B-INSTRUCT with the Italian prompt and GEMMA-3-12B-IT with the direct prompt, with scores of 0.897 and 0.895. When considering the complement of the MAE, both panels in the BINARY approach outperform all CONTINUOUS systems, achieving scores of 0.800 and 0.795. This likely reflects the noise-reducing effect of ensembling four LLMs, which lowers average error.

Overall, Panel 1 emerges as the best-performing system on the development set for the main task. This panel achieves $(1 - \text{MSE}) = 0.898$ and $(1 - \text{MAE}) = 0.800$, indicating that model predictions differ from the gold standard by approximately 0.20 units on average. The MSE of 0.10 further suggests that large errors are infrequent, as squared deviations remain low on average.

Few-shot Results In the *few-shot* setting, most models and all panels achieve similar or slightly lower performance in predicting the GS values compared to the *zero-shot* setting. In contrast, some models benefit from the inclusion of five examples. In particular, GEMMA-3-12B-IT and GEMMA-3-27B-IT for the CONTINUOUS approach show consistent improvements across prompts when evaluated in *few-shot*. Overall, the best-performing model-prompt combination in the *few-shot* setting is the CONTINUOUS GEMMA-3-27B-IT with the guidelines-informed prompt, achieving a complement of the MSE of 0.899. This configuration exhibits the best performance across *zero-shot* and *few-shot* according to the MSE.

Summary of Results These experiments on the development data show that both the CONTINUOUS and BINARY (panel-based) approaches can effectively predict the GS value. The three best-performing models for the CONTINUOUS approach: LLAMA-3.3-70B-INSTRUCT with guidelines-informed and Italian prompts, and GEMMA-3-12B-IT with the Italian prompt, together with the two panels for the BINARY approach (Table 2b), were selected for the five *zero-shot* runs in the main task on the official test data (see Section 5). Similarly, for the five *few-shot* runs, we selected the same two panels for the BINARY approach, along with the three best-performing models for the few-shot CONTINUOUS setting: GEMMA-3-27B-IT with a guidelines-informed prompt, and GEMMA-3-12B-IT with both the direct and Italian prompts.

4.3. GS Category Classification Results on Development Data (Subtask)

This experiment aims to evaluate *GS category* classification (subtask) on development data across all model-prompt combinations using the CONTINUOUS and the BINARY approaches.

Evaluation metrics We compute the macro precision, recall, and F1 score between the predicted and gold GS categories. The metrics are averaged across seven categories: the six stereotype categories: *role*, *personality*, *competence*, *physical*, *sexual*, and *relation*; and additionally include the *no* category.

Results Table 2c reports the macro F1, precision (P), and recall (R) for all model-prompt combinations for the CONTINUOUS and BINARY approaches, in *zero-shot* and *few-shot*. For the BINARY approach, unlike the computation of the GS value (which requires aggregating four binary predictions), each LLM directly predicts a stereotype category. Consequently, in this experiment, we evaluate the ability of each model-prompt pair to predict the category independently, including for the BINARY approach.³

Zero-shot Results In the *zero-shot* setting, models prompted for the BINARY approach tend to outperform those for the CONTINUOUS ones in predicting the GS category. This is likely because LLMs are conditioned to predict the category only when a stereotype is detected, and the continuous nature of the CONTINUOUS approach may introduce more noise, which can propagate to the category classification. We also observed results that contrasted with those for GS value (or binary label) prediction. In particular, the CoT prompt often outperforms other prompts in category classification, whereas it was the worst-performing prompt for the main task. This suggests that LLMs benefit more from reasoning through

³We also evaluated majority voting within panels for category prediction; however, this strategy resulted in lower performance.

chains of thought when performing category classification than when predicting binary stereotype labels or GS values. Surprisingly, the guidelines-informed prompt is among the best prompts for the CONTINUOUS approach but the worst for the BINARY approach in GS category classification.

The best-performing models are as follows: for the BINARY approach, GEMMA-3-27B with direct prompt (F1 = 0.56), GEMMA-3-12B with CoT prompt (F1 = 0.56), and LLAMA-3.3-70B-INSTRUCT with CoT prompt (F1 = 0.53); for the CONTINUOUS approach, QWEN3-8B with CoT prompt (F1 = 0.52), GEMMA-3-27B with guidelines-informed prompt (F1 = 0.51), and GEMMA-3-27B with Italian prompt (F1 = 0.50).

Few-shot Results Most model-prompt configurations within the CONTINUOUS approach show improved GS category classification performance in the *few-shot* setting, with the exception of LLAMA-3.3-70B-INSTRUCT, which exhibits a performance decrease in few-shot not only for category classification but also for the main task. We do not report quantitative results for the BINARY approach, as each model-prompt configuration yields different outcomes depending on the annotator. Nevertheless, qualitative inspection suggests that few-shot prompting does not lead to substantial improvements.

Summary of Results Although some model-prompt combinations in the CONTINUOUS approach performed better on GS *category* prediction than those selected for the main task (GS value prediction), we prioritized the main task. Therefore, we submitted GS category predictions from the same model-prompt combinations used for GS value prediction (Section 2a). Specifically, for the *zero-shot* setting, we used LLAMA-3.3-70B-INSTRUCT with guidelines-informed and Italian prompts (F1 = 0.49), and GEMMA-3-12B-IT with the Italian prompt (F1 = 0.45). For the *few-shot* setting, we selected GEMMA-3-27B-IT with a guidelines-informed prompt (F1 = 0.57), and GEMMA-3-12B-IT with both the direct (F1 = 0.46) and Italian prompts (F1 = 0.46). In contrast, for the BINARY approach, we selected the GS category predicted by GEMMA-3-27B with the direct prompt (F1 = 0.56), as this model-prompt combination achieved the highest overall performance in GS category prediction in the zero-shot setting and is present in both experimental panels. As a result, this model-prompt combination serves as the oracle for GS category prediction in both panels. Our final submitted systems are summarized in Table 3.

5. Results on the Test Data and Official Rankings

5.1. GS Value Prediction Results on Test Data (Main task)

Evaluation metrics We evaluate the performance in predicting the GS value using the normalized Mean Squared Error (nMSE), computed as the MSE normalized by the variance of the gold GS values: $\text{nMSE} = \text{MSE} / \text{Var}(y)$, where y are the ground-truth GS values. Systems are ranked using the transformed score $1/(1 + \text{nMSE})$ so that higher values correspond to better performance.

Results Table 4 reports the official rankings for the main task for *zero-shot* and *few-shot* settings.

Zero-shot Results In the zero-shot setting, our best submission ranked 2nd overall with a score of 0.626. This system used GEMMA-3-12B with the guidelines-informed prompt under the CONTINUOUS approach. Our second-best system, LLAMA-3.3-70B-INSTRUCT with the Italian prompt, ranked 3rd with a score of 0.622. The top-performing system from team DIAG-Sapienza achieved a score of 0.700 using GPT-5. Notably, all three of our CONTINUOUS submissions (ranks 2, 3, and 6) outperformed the two BINARY panel submissions (ranks 11 and 12). This contrasts with our development set findings, where the panels achieved competitive performance. The gap between the CONTINUOUS and BINARY approaches on the test set may indicate that the panel compositions, optimized on the development data, did not generalize well to the test data distribution.

Few-shot Results In the few-shot setting, our best submission again ranked 2nd (score of 0.645). This system used GEMMA-3-27B with the guidelines-informed prompt. The GEMMA-3-12B model with the direct prompt ranked 3rd (0.629), and with the Italian prompt ranked 4th (0.612). All of these systems follow the CONTINUOUS approach. Compared to the zero-shot setting, our few-shot submissions showed some but modest improvements, with the best score increasing from 0.626 to 0.645. The BINARY Panel 2 (F1-based) achieved a score of 0.586 (rank 6), while Panel 1 (Cohen’s κ -based) dropped to 0.549 (rank 13). This suggests that the few-shot examples may have introduced noise for certain panel configurations.

Table 2

Results on the development data. Results are reported for the GS value prediction (a) and GS category classification (c) tasks. For GS value prediction (a), the complement of nMSE and nMAE is reported for all models under the CONTINUOUS approach and for the two panels of the BINARY approach, summarized in (c). For GS category classification (c), the F1 score, Precision (P), and Recall (R) are reported for all model-prompt combinations under both approaches. Model-prompt combinations are grouped by approach, with *Zero-Shot* and *Few-Shot* results presented side by side. Bold values indicate the top-performing configurations. The scores of the five submitted configurations, separately for zero-shot and few-shot settings, are underlined and highlighted.

(a) GS Value Prediction (Main Task)					(c) GS Category Classification (Subtask)								
Model	Prompt	Zero-Shot		Few-Shot		Model	Prompt	Zero-Shot			Few-Shot		
		(1-MSE) ↑	(1-MAE) ↑	(1-MSE) ↑	(1-MAE) ↑			F1	P	R	F1	P	R
Continuous Approach					Continuous Approach								
Llama-3.1-8B-Instruct	Direct	.849	.713	.856	.725	Llama-3.1-8B-Instruct	Direct	.25	.44	.28	.33	.55	.34
	CoT	.737	.618	.778	.651		CoT	.42	.48	.47	.41	.46	.45
	Guidelines	.832	.658	.819	.660		Guidelines	.37	.44	.40	.40	.42	.46
	Italian	.830	.687	.791	.655		Italian	.29	.39	.32	.30	.32	.33
	Perspectivist	.826	.671	.826	.690		Perspectivist	.28	.34	.31	.31	.30	.36
Llama-3.3-70B-Instruct	Direct	.888	.743	.573	.493	Llama-3.3-70B-Instruct	Direct	.44	.50	.47	.04	.02	.14
	CoT	.879	.739	.573	.493		CoT	.48	.50	.54	.04	.02	.14
	Guidelines	.898	.752	.573	.493		Guidelines	.49	.50	.53	.04	.02	.14
	Italian	.897	.764	.573	.493		Italian	.49	.51	.52	.04	.02	.14
	Perspectivist	.867	.741	.573	.493		Perspectivist	.39	.48	.45	.04	.02	.14
gemma-3-4b-it	Direct	.834	.650	.861	.682	gemma-3-4B	Direct	.31	.54	.37	.33	.45	.40
	CoT	.769	.618	.824	.648		CoT	.33	.43	.41	.33	.40	.40
	Guidelines	.805	.632	.843	.667		Guidelines	.38	.40	.46	.40	.49	.47
	Italian	.852	.676	.815	.645		Italian	.33	.44	.41	.29	.38	.38
	Perspectivist	.727	.604	.788	.649		Perspectivist	.18	.41	.23	.34	.35	.37
gemma-3-12b-it	Direct	.895	.745	.895	.751	gemma-3-12B	Direct	.40	.50	.42	.46	.53	.48
	CoT	.841	.686	.881	.730		CoT	.48	.53	.54	.52	.54	.57
	Guidelines	.854	.687	.881	.730		Guidelines	.45	.49	.52	.51	.53	.56
	Italian	.898	.757	.894	.760		Italian	.45	.55	.47	.43	.49	.46
	Perspectivist	.860	.719	.870	.728		Perspectivist	.39	.48	.44	.44	.45	.45
gemma-3-27b-it	Direct	.891	.745	.891	.757	gemma-3-27B	Direct	.44	.49	.49	.51	.52	.54
	CoT	.885	.739	.887	.768		CoT	.47	.50	.53	.57	.59	.60
	Guidelines	.869	.715	.899	.763		Guidelines	.51	.55	.60	.57	.57	.63
	Italian	.885	.731	.886	.751		Italian	.50	.52	.55	.53	.56	.56
	Perspectivist	.831	.688	.867	.714		Perspectivist	.41	.53	.51	.46	.46	.48
Qwen3-8B	Direct	.862	.716	.850	.726	Qwen3-8B	Direct	.40	.47	.41	.49	.59	.50
	CoT	.797	.696	.802	.701		CoT	.52	.54	.54	.55	.58	.57
	Guidelines	.852	.739	.859	.744		Guidelines	.49	.49	.51	.54	.55	.55
	Italian	.848	.721	.816	.690		Italian	.46	.47	.47	.46	.51	.47
	Perspectivist	.796	.675	.821	.690		Perspectivist	.39	.45	.41	.39	.41	.43
Qwen3-32B	Direct	.875	.734	.875	.755	Qwen3-32B	Direct	.44	.49	.47	.53	.57	.55
	CoT	.810	.700	.830	.729		CoT	.47	.50	.51	.54	.55	.57
	Guidelines	.879	.744	.862	.747		Guidelines	.54	.54	.56	.55	.55	.57
	Italian	.886	.748	.853	.727		Italian	.49	.53	.52	.48	.50	.51
	Perspectivist	.869	.738	.869	.736		Perspectivist	.42	.49	.46	.48	.53	.51
LLaMAntino-3-ANITA-8B-Inst-DPO-ITA	Direct	.876	.729	.861	.720	LLaMAntino-3-ANITA-8B-Inst-DPO-ITA	Direct	.31	.48	.32	.35	.51	.37
	CoT	.820	.692	.829	.703		CoT	.39	.44	.40	.40	.44	.42
	Guidelines	.813	.664	.797	.653		Guidelines	.34	.38	.38	.34	.34	.39
	Italian	.871	.720	.723	.603		Italian	.31	.49	.33	.29	.32	.34
	Perspectivist	.855	.743	.835	.705		Perspectivist	.27	.41	.30	.36	.42	.38
Binary Approach					Binary Approach								
Panel 1 (Cohen's κ)	—	.898	.800	.882	.794	Llama-3.1-8B-Instruct	Direct	.35	.55	.35	—	—	—
Panel 2 (F1)	—	.883	.795	.868	.760		CoT	.43	.49	.45	—	—	—
							Guidelines	.13	.11	.22	—	—	—
							Italian	.37	.47	.37	—	—	—
						Llama-3.3-70B-Instruct	Direct	.47	.56	.46	—	—	—
							CoT	.53	.55	.55	—	—	—
							Guidelines	.15	.12	.24	—	—	—
							Italian	.45	.62	.43	—	—	—
						gemma-3-4B	Direct	.33	.56	.36	—	—	—
							CoT	.42	.50	.51	—	—	—
							Guidelines	.14	.11	.23	—	—	—
							Italian	.35	.61	.40	—	—	—
						gemma-3-12B	Direct	.33	.56	.36	—	—	—
							CoT	.56	.57	.57	—	—	—
							Guidelines	.14	.12	.23	—	—	—
							Italian	.47	.63	.49	—	—	—
						gemma-3-27B	Direct (Oracle [P1] - [P2])	.56	.59	.56	—	—	—
							CoT	.55	.56	.56	—	—	—
							Guidelines	.14	.12	.23	—	—	—
							Italian	.48	.52	.50	—	—	—
						Qwen3-8B	Direct	.33	.54	.32	—	—	—
							CoT	.41	.63	.37	—	—	—
							Guidelines	.11	.22	.17	—	—	—
							Italian	.30	.55	.30	—	—	—
						Qwen3-32B	Direct	.37	.50	.36	—	—	—
							CoT	.52	.58	.52	—	—	—
							Guidelines	.20	.34	.25	—	—	—
							Italian	.39	.61	.38	—	—	—
						LLaMAntino-3-ANITA-8B-Inst-DPO-ITA	Direct	.32	.52	.34	—	—	—
							CoT	.39	.53	.39	—	—	—
							Guidelines	.15	.11	.24	—	—	—
							Italian	.31	.55	.33	—	—	—

(b) Panels composition

P1 - Panel (Cohen's κ)

Annotator	Model	Prompt	k
A1	Llama-3.3-70B-Instruct	Italian	0.51
A2	Llama-3.1-8B-Instruct	Direct	0.50
A3	LLaMAntino-3-ANITA-8B	Guidelines	0.52
A4	gemma-3-27b-it	Direct	0.57

P2 - Panel (F1)

Annotator	Model	Prompt	F1
A1	Llama-3.3-70B-Instruct	Direct	0.75
A2	Llama-3.3-70B-Instruct	Italian	0.75
A3	gemma-3-12b-it	Direct	0.76
A4	gemma-3-27b-it	Direct	0.78

(b) Panels composition

P1 - Panel (Cohen's κ)			
Annotator	Model	Prompt	k
A1	Llama-3.3-70B-Instruct	Italian	0.51
A2	Llama-3.1-8B-Instruct	Direct	0.50
A3	LLaMAntino-3-ANITA-8B	Guidelines	0.52
A4	gemma-3-27b-it	Direct	0.57
P2 - Panel (F1)			
Annotator	Model	Prompt	F1
A1	Llama-3.3-70B-Instruct	Direct	0.75
A2	Llama-3.3-70B-Instruct	Italian	0.75
A3	gemma-3-12b-it	Direct	0.76
A4	gemma-3-27b-it	Direct	0.78

Table 3
Summary of Submitted Systems.

Run	Model ID	Config ID	Setting	Approach	GS Value Prediction	GS Category Classification
<i>Zero-Shot Submissions</i>						
1	Panel_4LLMS	K_Cohen	Zero-shot	Binary	Panel 1: Aggregation of 4 binary labels	gemma-3-27b-it (Direct)
2	Panel_4LLMS	F1_score	Zero-shot	Binary	Panel 2: Aggregation of 4 binary labels	gemma-3-27b-it (Direct)
3	Llama_3.3_70B	P3GUIDE	Zero-shot	Continuous	Llama-3.3-70B-Instruct (Guidelines)	Llama-3.3-70B-Instruct (Guidelines)
4	Llama_3.3_70B	P5ITA	Zero-shot	Continuous	Llama-3.3-70B-Instruct (Italian)	Llama-3.3-70B-Instruct (Italian)
5	Gemma_3_12B	P5ITA	Zero-shot	Continuous	gemma-3-12b-it (Italian)	gemma-3-12b-it (Italian)
<i>Few-Shot Submissions</i>						
1	Panel_4LLMS	K_Cohen	Few-shot	Binary	Panel 1: Aggregation of 4 binary labels	gemma-3-27b-it (Direct)
2	Panel_4LLMS	F1_score	Few-shot	Binary	Panel 2: Aggregation of 4 binary labels	gemma-3-27b-it (Direct)
3	Gemma_3_27B	P3GUIDE	Few-shot	Continuous	gemma-3-27b-it (Guidelines)	gemma-3-27b-it (Guidelines)
4	Gemma_3_12B	P1DIR	Few-shot	Continuous	gemma-3-12b-it (Direct)	gemma-3-12b-it (Direct)
5	Gemma_3_12B	P5ITA	Few-shot	Continuous	gemma-3-12b-it (Italian)	gemma-3-12b-it (Italian)

Summary of Results Our systems achieved strong performance in the main task, ranking best among open-weight models, and overall, 2nd in both zero-shot and few-shot settings. The CONTINUOUS approach outperformed the BINARY approach on the test data. The guidelines-informed prompt proved effective across both settings and model sizes. These results suggest directly predicting the aggregated GS value is more robust than modeling individual annotators when generalizing to unseen data.

5.2. GS Category Classification Results on Test Data (Subtask)

Evaluation metrics We evaluate performance on GS category classification using the micro F1 score as the primary metric, following the official evaluation guidelines. We also report the macro F1 score.⁴

Results Table 5 reports the official rankings for the subtask, separately for *zero-shot* and *few-shot*.

Zero-shot Results In zero-shot, our submissions achieved the top four positions in the leaderboard. The best-performing system was Panel 2 (F1-based) with GEMMA-3-27B as the category oracle, achieving a micro F1 of 0.646. Panel 1 (Cohen’s κ -based) ranked 3rd with a micro F1 of 0.643. The BINARY panels outperformed the CONTINUOUS systems for category classification, despite showing the opposite trend for GS value prediction. This indicates that the oracle model (GEMMA-3-27B with the direct prompt) used for category prediction in the panels was highly effective. The CONTINUOUS submissions using GEMMA-3-12B and LLAMA-3.3-70B ranked 2nd and 4th, with F1 micro of 0.644 and 0.634.

Few-shot Results In the few-shot setting, our best submission ranked 3rd with a micro F1 of 0.669. This system used GEMMA-3-27B with the guidelines-informed prompt under the CONTINUOUS approach. The top two positions were held by teams Prisma (0.706) and FestaDonato (0.671) using Claude and Gemini models, respectively. Both BINARY panels ranked 5th and 6th with identical micro F1 scores of 0.662. The remaining CONTINUOUS submissions (GEMMA-3-12B with direct and Italian prompts) ranked 11th and 12th. This drop in performance for these configurations suggests that the few-shot examples were less beneficial for smaller models in the category classification task. Similarly to the main task, our models ranked first in both settings, considering only non-commercial, open-weight models.

Summary of Results Our systems achieved 1st place in the zero-shot and 3rd place in the few-shot settings for category classification. In the zero-shot setting, the BINARY panels performed best, likely due to the strong category prediction ability of GEMMA-3-27B (direct prompt), which was used as the sole category predictor for both panels. In the few-shot setting, larger models with the guidelines-informed prompt performed best. Overall, category classification benefited from explicit reasoning about stereotype types, as reflected in the strong performance of the guidelines-informed prompt.

6. Conclusion

In this paper, we described our contribution to the GSI:DETECT task on stereotype detection at EVALITA 2026. We have presented two different LLM-based approaches: (1) directly predicting the degree of stereotype (CONTINUOUS), and (2) predicting the binary presence of gender stereotypes and aggregating multiple individual predictions to compute the degree of stereotype (BINARY). We also have proposed

⁴Official rankings F1 scores only over the six GS categories (*role, personality, competence, physical, sexual, relational*), excluding instances with no stereotype. This differs from our development analysis, which included the *no stereotype* category.

Table 4

Official GSI:detect leaderboard for **GS value** prediction (**Main Task**), separately for *zero-shot* (a) and *few-shot* (b).

	1/1+nMSE	team_name	model	configuration
1	0.700	DIAG-Sapienza	GPT-5	gen_with_explanation
2	0.626	StereoBusters	Llama_3.3_70B	P3GUIDE
3	0.622	StereoBusters	Llama_3.3_70B	P5ITA
4	0.621	FestaDonato	Gemini_2.5_Flash	engNegative_V2
5	0.618	FestaDonato	Gemini_2.5_Flash	ensemble
6	0.610	StereoBusters	Gemma_3_12B	P5ITA
7	0.610	FestaDonato	Gemini_2.5_Flash	ITA_DIR-V3
8	0.609	BASELINE	GPT-5 nano	split_prompt
9	0.591	Tiz	Gemma_3_12B	average
10	0.590	BASELINE	GPT-5 nano	unified_prompt
11	0.570	StereoBusters	Panel_4LLMs	F1_score
12	0.565	StereoBusters	Panel_4LLMs	K_Cohen
13	0.557	FestaDonato	Gemini_2.5_Flash	CoT-eng_V5
14	0.555	FestaDonato	Gemini_2.5_Flash	CoT-ita_V4
15	0.543	BASELINE	Qwen-3_14B	unified_prompt
16	0.498	BASELINE	N/A	GS_value0.5
17	0.484	Tiz	Gemma_3_12B	max3
18	0.483	Tiz	Gemma_3_12B	max2
19	0.472	Tiz	Gemma_3_12B	max1
20	0.430	Tiz	Gemma_3_12B	all_zeros
21	0.329	Prisma	Claude_3.5_Sonnet	persona_A
22	0.321	Prisma	Claude_3.5_Sonnet	persona_C
23	0.321	Prisma	Claude_3.5_Sonnet	4_annotators
24	0.309	Prisma	Claude_3.5_Sonnet	persona_D
25	0.309	Prisma	Claude_3.5_Sonnet	persona_B

(a) Zero-shot setting

	1/1+nMSE	team_name	model	configuration
1	0.685	DIAG-Sapienza	GPT-5	gen_with_explanation
2	0.645	StereoBusters	Gemma_3_27B	P3GUIDE
3	0.629	StereoBusters	Gemma_3_12B	P1DIR
4	0.612	StereoBusters	Gemma_3_12B	P5ITA
5	0.587	Prisma	Claude_3.5_Sonnet	stratified_sampling
6	0.586	StereoBusters	Panel_4LLMs	F1_score
7	0.581	MINDS	Qwen2.5_14B	KNN
8	0.572	FestaDonato	Gemini_2.5_Flash	Ensemble
9	0.565	FestaDonato	Gemini_2.5_Flash	Semantic_Clustering
10	0.564	MINDS	Qwen2.5_14B	LR
11	0.561	FestaDonato	Gemini_2.5_Flash	Semantic_Clustering
12	0.550	FestaDonato	Gemini_2.5_Flash	Hybrid
13	0.549	StereoBusters	Panel_4LLMs	K_Cohen
14	0.500	FestaDonato	Gemini_2.5_Flash	Manual

(b) Few-shot setting

Table 5

Official GSI:detect leaderboard for **GS category** classification (**SubTask**), for *zero-shot* (a) and *few-shot* (b).

	F1 Micro	F1 Macro	team_name	model	configuration
1	0.646	0.637	StereoBusters	Panel_4LLMs	F1_score
2	0.644	0.637	StereoBusters	Llama_3.3_70B	P3GUIDE
3	0.643	0.633	StereoBusters	Panel_4LLMs	K_Cohen
4	0.634	0.630	StereoBusters	Llama_3.3_70B	P5ITA
5	0.600	0.561	FestaDonato	Gemini_2.5_Flash	CoT-eng_V5
6	0.597	0.552	FestaDonato	Gemini_2.5_Flash	ensemble
7	0.590	0.552	FestaDonato	Gemini_2.5_Flash	engNegative_V2
8	0.586	0.543	FestaDonato	Gemini_2.5_Flash	ITA_DIR-V3
9	0.581	0.519	DIAG-Sapienza	GPT-5	gen_with_explanation
10	0.569	0.523	FestaDonato	Gemini_2.5_Flash	CoT-ita_V4
11	0.539	0.534	StereoBusters	Gemma_3_12B	P5ITA
12	0.532	0.519	BASELINE	GPT-5 nano	split_prompt
13	0.516	0.504	BASELINE	GPT-5 nano	unified_prompt
14	0.394	0.386	BASELINE	Qwen-3_14B	unified_prompt
15	0.229	0.229	Tiz	Gemma_3_12B	max2
16	0.229	0.229	Tiz	Gemma_3_12B	max3
17	0.224	0.223	Tiz	Gemma_3_12B	max1
18	0.224	0.290	Prisma	Claude_3.5_Sonnet	4_annotators
19	0.222	0.222	Tiz	Gemma_3_12B	average
20	0.208	0.208	Tiz	Gemma_3_12B	all_zeros
21	0.181	0.179	BASELINE	N/A	GS_category: random
22	0.180	0.229	Prisma	Claude_3.5_Sonnet	persona_C
23	0.174	0.236	Prisma	Claude_3.5_Sonnet	persona_A
24	0.129	0.178	Prisma	Claude_3.5_Sonnet	persona_B
25	0.122	0.171	Prisma	Claude_3.5_Sonnet	persona_D

(a) Zero-shot setting

	F1 Micro	F1 Macro	team_name	model	configuration
1	0.706	0.612	Prisma	Claude_3.5_Sonnet	stratified_sampling
2	0.671	0.669	FestaDonato	Gemini_2.5_Flash	Hybrid
3	0.669	0.665	StereoBusters	Gemma_3_27B	P3GUIDE
4	0.666	0.656	FestaDonato	Gemini_2.5_Flash	Ensemble
5	0.662	0.653	StereoBusters	Panel_4LLMs	K_Cohen
6	0.662	0.654	StereoBusters	Panel_4LLMs	F1_score
7	0.657	0.653	FestaDonato	Gemini_2.5_Flash	Semantic_Clustering
8	0.650	0.635	FestaDonato	Gemini_2.5_Flash	Semantic_Clustering
9	0.636	0.627	FestaDonato	Gemini_2.5_Flash	Manual
10	0.606	0.549	DIAG-Sapienza	GPT-5	gen_with_explanation
11	0.574	0.572	StereoBusters	Gemma_3_12B	P1DIR
12	0.565	0.561	StereoBusters	Gemma_3_12B	P5ITA

(b) Few-shot setting

different prompting strategies and evaluated how different LLMs perform for each of these. Finally, we conducted an additional annotation experiment using the annotation guidelines (see Appendix C).

Our results show that open-weight models, even with a relatively small number of parameters (e.g., Gemma 12B), can be highly effective, when properly prompted, at detecting both the presence and type of stereotypes, achieving performance close to that of more powerful commercial models. We found that both proposed approaches are promising for identifying and classifying gender stereotypes. However, while the BINARY approach outperformed the CONTINUOUS approach on the development data, it generalized less effectively to the test data. We also observed that few-shot prompting tends to yield limited performance improvements, except for the CONTINUOUS approach on the main task.

In general, all model-prompt combinations showed good performance, the main exception being the CoT prompt, which tends to exhibit lower performance for the main task. This pattern was partially flipped in the case of the subtask, where the CoT model showed competitive performance when compared to the other prompting strategies, if not straight outperforming them.

Acknowledgments

The work of M. Marchiori Manerba is partially funded by the project CSP TRAPEZIO 2025-2025.0998 “Neikea-HS: NLP and Ethical AI for Knowledge-based Education Against Hate Speech”. The work of A.T. Cignarella is supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, Grant Agreement No. 101146287. Computational resources were provided by King’s Computational Research, Engineering and Technology Environment (CREATE) at King’s College London [32].

Declaration on Generative AI

During the preparation of this work, the authors used **ChatGPT** and **Grammarly** to: **Grammar and spelling check, and Paraphrase and reword**. The authors reviewed the content and take full responsibility for it.

References

- [1] C. Bosco, F. Dell’Orletta, F. Poletto, M. Sanguinetti, M. Tesconi, Overview of the EVALITA 2018 Hate Speech Detection Task, in: Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018, volume 2263 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2018.
- [2] M. Sanguinetti, G. Comandini, E. D. Nuovo, S. Frenda, M. Stranisci, C. Bosco, T. Caselli, V. Patti, I. Russo, HaSpeeDe 2 @ EVALITA2020: Overview of the EVALITA 2020 Hate Speech Detection Task, in: V. Basile, D. Croce, M. D. Maro, L. C. Passaro (Eds.), Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), volume 2765 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020.
- [3] E. Fersini, D. Nozza, P. Rosso, Overview of the evalita 2018 task on automatic misogyny identification (AMI), in: Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR Workshop Proceedings, 2018.
- [4] E. Fersini, D. Nozza, P. Rosso, AMI @ EVALITA2020: automatic misogyny identification, in: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), 2020.
- [5] D. Nozza, A. T. Cignarella, G. Damo, T. Caselli, V. Patti, HODI at EVALITA 2023: Overview of the first Shared Task on Homotransphobia Detection in Italian, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023.
- [6] M. Nissim, D. Croce, V. Patti, P. Basile, G. Attanasio, et al., Challenging the abilities of large language models in italian: a community initiative, 2025. URL: <https://arxiv.org/abs/2512.04759>.
- [7] C. Bosco, V. Patti, S. Frenda, A. T. Cignarella, M. Paciello, F. D’Errico, Detecting racial stereotypes: An italian social media corpus where psychology meets nlp, *Information Processing Management* 60 (2023) 103118. doi:<https://doi.org/10.1016/j.ipm.2022.103118>.
- [8] W. Wolfgang Schmeisser-Nieto, G. Ricci, S. Frenda, M. Taule, C. Bosco, Implicit stereotypes: A corpus-based study for Italian, in: Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 997–1004. URL: <https://aclanthology.org/2024.clicit-1.108/>.
- [9] B. Cristina, P. Marinella, F. Benamara, C. P. Giovanni, P. Viviana, M. Véronique, T. Mariona, et al., Sterheotypes project. detecting and countering ethnic stereotypes emerging from italian, spanish and french racial hoaxes, in: Proceedings of the Seminar of the Spanish Society for Natural Language Processing: Projects and System Demonstrations (SEPLN-CEDI-PD 2024), 2024.
- [10] M. Berta, S. Greco, G. Tipaldo, T. Cerquitelli, Decoding narratives: Towards a classification analysis for stereotypical patterns in italian news headlines, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 5253–5262. doi:10.1109/BigData62323.2024.10825258.
- [11] S. Greco, M. La Quatra, L. Cagliero, T. Cerquitelli, Towards ai-assisted inclusive language writing in italian formal communications, *ACM Trans. Intell. Syst. Technol.* 16 (2025). URL: <https://doi.org/10.1145/3729237>. doi:10.1145/3729237.
- [12] B. Savoldi, G. Attanasio, E. Cupin, E. Gkovedarou, J. Hackenbuchner, A. Lauscher, M. Negri, A. Piergentili, M. Thind, L. Bentivogli, Mind the inclusivity gap: Multilingual gender-neutral translation evaluation with mGeNTE, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2025, pp. 13709–13731. doi:10.18653/v1/2025.emnlp-main.692.
- [13] M. La Quatra, S. Greco, L. Cagliero, T. Cerquitelli, Inclusively: An ai-based assistant for inclusive writing, in: Machine Learning and Knowledge Discovery in Databases: Applied Data Science

- and Demo Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part VII, 2023, p. 361–365. doi:10.1007/978-3-031-43430-3_31.
- [14] M. L. Quatra, S. Greco, L. Cagliero, M. Tonti, F. Dragotto, R. Raus, S. Cavagnoli, T. Cerquitelli, Building foundations for inclusiveness through expert-annotated data, in: EDBT/ICDT Workshops, 2024. URL: <https://ceur-ws.org/Vol-3651/DARLI-AP-3.pdf>.
 - [15] G. Attanasio, S. Greco, M. La Quatra, L. Cagliero, M. Tonti, T. Cerquitelli, R. Raus, E-mimic: Empowering multilingual inclusive communication, in: 2021 IEEE International Conference on Big Data (Big Data), 2021, pp. 4227–4234. doi:10.1109/BigData52589.2021.9671868.
 - [16] A. Piergentili, B. Savoldi, M. Negri, L. Bentivogli, Gender-neutral rewriting in Italian: Models, approaches, and trade-offs, in: Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 899–910. URL: <https://aclanthology.org/2025.clicit-1.84/>.
 - [17] K. Stanczak, I. Augenstein, A Survey on Gender Bias in Natural Language Processing, arXiv preprint arXiv:2112.14168 (2021).
 - [18] A. T. Cignarella, A. Giachanou, E. Lefever, A Survey on Stereotype Detection in Natural Language Processing, ACM Computing Surveys 58 (2025) 1–33. doi:10.1145/3770754.
 - [19] G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini, Gsi: detect at evalita 2026: Overview of the task on detecting gender stereotypes in italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [20] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [21] M. Sanguinetti, F. Poletto, C. Bosco, V. Patti, M. Stranisci, An Italian Twitter corpus of hate speech against immigrants, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: <https://aclanthology.org/L18-1443/>.
 - [22] V. Basile, et al., It’s the end of the gold standard as we know it. on the impact of pre-aggregation on the evaluation of highly subjective tasks, volume 2776, CEUR-WS, 2020, pp. 31–40.
 - [23] F. Cabitza, A. Campagner, V. Basile, Toward a perspectivist turn in ground truthing for predictive computing, Proceedings of the AAAI Conference on Artificial Intelligence 37 (2023) 6860–6868. doi:10.1609/aaai.v37i6.25840.
 - [24] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS ’22, 2022.
 - [25] A. Grattafiori, A. Dubey, A. Jauhri, et al., The llama 3 herd of models, 2024. arXiv:2407.21783.
 - [26] AI@Meta, Llama 3 model card (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.
 - [27] Q. Team, Qwen3 technical report, 2025. URL: <https://arxiv.org/abs/2505.09388>.
 - [28] M. Polignano, P. Basile, G. Semeraro, Advanced natural-based interaction for the italian language: Llamantino-3-anita, 2024. arXiv:2405.07101.
 - [29] P. Basile, E. Musacchio, M. Polignano, L. Siciliani, G. Fiameni, G. Semeraro, Llamantino: Llama 2 models for effective text generation in italian language, 2023. arXiv:2312.09993.
 - [30] G. Team, Gemma 3 (2025). URL: <https://goo.gle/Gemma3Report>.
 - [31] J. Cohen, A coefficient of agreement for nominal scales, Educational and psychological measurement 20 (1960) 37–46.
 - [32] King’s College London, King’s computational research, engineering and technology environment (create), <https://doi.org/10.18742/rnvf-m076>, 2025. Retrieved December, 2025.
 - [33] M. Sap, S. Gabriel, L. Qin, D. Jurafsky, N. A. Smith, Y. Choi, Social bias frames: Reasoning about social and power implications of language, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. doi:10.18653/v1/2020.acl-main.486.

Appendix

The appendix is organised as follows: Appendix A presents additional analysis of the GSI:DETECT dataset; Appendix B provides the LLMs’ prompts under the CONTINUOUS and BINARY approaches; and Appendix C discusses an additional human annotation experiment and a qualitative analysis of the data.

A. Analysis of the GSI:detect dataset

As introduced in Section 2, the GSI:DETECT dataset comprises 1,010 short Italian texts collected from both social media and information websites, covering both formal and informal registers. The dataset is split into 200 instances for development and 810 instances for testing. The information and annotations provided for each text are summarised in Table 6.

The annotation process involved four annotators with expertise and sensitivity to gender-related issues. Demographic characteristics of the annotators were released midway through the development phase of the task (see Table 7) to support the design of systems that better reflect annotator diversity.

The distribution of the *GS value* across the dataset is shown in Figure 1a. The development and test sets present similar distributions: approximately one third of the texts have a GS value of 0 (i.e., none of the annotators identified stereotypes), and approximately one third have a GS value of 1 (i.e., all annotators identified stereotypes). However, the test set contains slightly more texts with a GS value of 1, while the development set contains fewer texts with intermediate values (0.25 and 0.5).

Similarly, the distributions of the *GS category* follow a comparable pattern in the development and test sets, as shown in Figure 1b. The most common category other than “no stereotype” is the *competence* stereotype category, followed closely by *role* and *personality* stereotypes. The least represented categories are *sexual* and *relational* stereotypes. Although the overall distributions are similar across the two splits, *competence*, *role*, and *personality* stereotypes are more prevalent in the development set, whereas *physical*, *sexual*, and *relational* stereotypes occur more frequently in the test set.

We computed Cohen’s κ to quantify the agreement between annotators. Table 8 reports the pairwise agreement on the presence of stereotypes (i.e., yes/no label) for the development set, while Table 9 reports the agreement for the test set. The values for the development dataset lie between 0.45 and 0.63, which indicates moderate to substantial agreement. With the exception of two of the annotators (A1–A3), the values for the test set lie between 0.61 and 0.69, which indicates substantial agreement.

B. LLM Prompts

As discussed in Section 3, we proposed and evaluated five prompting strategies: (1) direct instructions, (2) integration of detailed guidelines, (3) Chain-of-Thought prompting, (4) use of Italian prompts, and (5) modeling the perspectives of different annotators. LLMs’ prompts are reported in Figure 2 for the CONTINUOUS approach and in Figure 3 for the BINARY approach.

For the *few-shot* setting, the prompt structure is identical across the two settings; however, in the *few-shot* setting, five illustrative examples are appended to the end of each prompt, as shown in the red text in Figure 2f. We report only this example, as all other prompts follow the same structure.

We designed all prompts to predict either the GS value or the binary gender stereotype label, together with the GS category. Models are instructed to assign the “no” category only when the GS value or label is 0 (i.e., when no stereotypes are detected).

Table 6

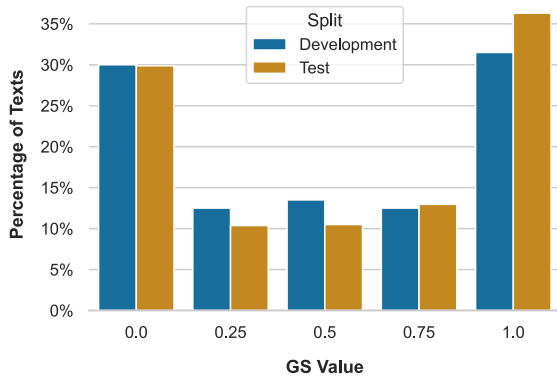
List of variables and annotations provided for each text within the GSI:detected dataset.

Variable	Description
Text	The short text to be used for the task and any additional context.
Annotations	A list containing four binary annotations denoting the presence of stereotypes in the text, one entry for each annotator.
GS Value	The GS value for the text. As mentioned in Section ??, this is a numerical value between 1 and 0.
GS Category	The GS category for the text.
Context	A binary label denoting whether additional information regarding the context in which the text appears.
Annotations	A list containing four binary annotations (yes/no) about the presence of stereotypes in the text, one for each annotator.

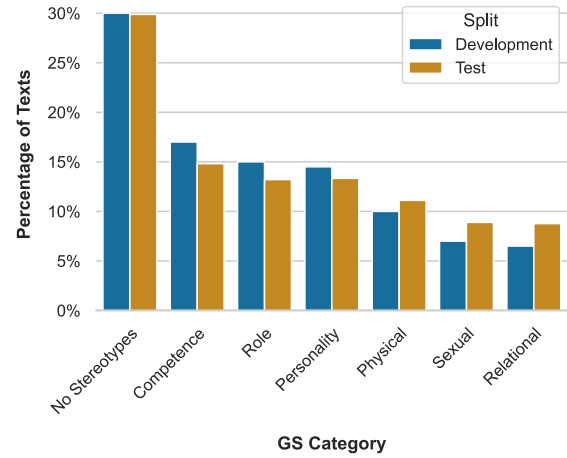
Table 7

Demographic characteristics of the four annotators.

Characteristic	Value	Count
Gender	Cis woman	3
	Cis man	1
Age	20-30	2
	30-40	1
	40+	1
Educational Level	PhD	1
	Masters	3
First Language	Italian	4



(a) GS values distribution



(b) GS categories distribution (sorted by frequency)

Figure 1: Distribution of the GS values and categories across the development and test sets.

Table 8

Pairwise agreement of the development set.

Ann_A	Ann_B	Cohen's κ
A1	A2	0.6172
A1	A3	0.4528
A1	A4	0.5527
A2	A3	0.5603
A2	A4	0.6203
A3	A4	0.6328

Table 9

Pairwise agreement of the test set.

Ann_A	Ann_B	Cohen's κ
A1	A2	0.6250
A1	A3	0.4972
A1	A4	0.6534
A2	A3	0.6107
A2	A4	0.6912
A3	A4	0.6772

```

system: |
You are an expert in detecting gender stereotypes in Italian text.
Your task is to analyze texts and provide two outputs:
1. A GS value (0.0 to 1.0) indicating the degree of gender stereotype present
2. The category of stereotype if present

GS Value Scale:
- 0.0 = No stereotype
- 0.25 = Weak stereotype (1 of 4 annotators would identify it)
- 0.50 = Moderate stereotype (2 of 4 annotators would identify it)
- 0.75 = Strong stereotype (3 of 4 annotators would identify it)
- 1.0 = Very strong stereotype (all annotators would identify it)

Categories:
- role: Social/cultural expectations about what men/women should do
- personality: Emotional/behavioral traits assigned by gender
- competence: Judgments about abilities based on gender
- physical: Expectations about appearance and body
- sexual: Attitudes about sexuality and sexual behavior
- relational: How genders should behave in relationships
- no: No stereotype present

IMPORTANT: Respond ONLY with the JSON output. Do not include any explanations
before or after.
Format your response EXACTLY as: JSON_OUTPUT: {"gs_value": X.XX, "gs_category":
"category"}

user: |
Analyze this Italian text for gender stereotypes:

Text: "{text}"

Provide your analysis as JSON_OUTPUT: {"gs_value": <number between 0.0 and 1.0>,
"gs_category": "<role|personality|competence|physical|sexual|relational|no>"}
user_with_context: |
Analyze this Italian text for gender stereotypes. The text includes context
that should inform your evaluation.

Context: "{context}"

Text to evaluate: "{text}"

IMPORTANT: Evaluate the main text considering the provided context. The context
helps understand the situation but your analysis should focus on the main
text.

Provide your analysis as JSON_OUTPUT: {"gs_value": <number between 0.0 and 1.0>,
"gs_category": "<role|personality|competence|physical|sexual|relational|no>"}

```

(a) Zero-Shot Prompt 1 (CONTINUOUS): Direct

```

system: |
You are an expert annotator trained in detecting gender stereotypes according
to specific guidelines.

DEFINITION: Gender stereotypes are rigid and generalized beliefs about roles,
behaviors, abilities,
or characteristics that men and women "should" have based on their sex or
gender.

DETAILED CATEGORIES:

1. ROLE STEREOTYPES
- Men: breadwinners, heads of household, financial support, protection
- Women: mothers, housewives, child-rearing, domestic work
- Note: Work aspects usually fall under "competence"

2. PERSONALITY STEREOTYPES
- Women: empathetic, sensitive, emotional, indecisive, caring
- Men: strong, assertive, rational, confident, unemotional
- Includes gendered insults (e.g., "femminuccia", "avere le palle")

3. COMPETENCE STEREOTYPES
- Women: suited for humanities, supportive roles
- Men: skilled in STEM, logic, technology, leadership
- Includes sports, games, professional abilities

4. PHYSICAL STEREOTYPES
- Women: slim, young, attractive, well-groomed (judged harshly)
- Men: robust, can be scruffy (more accepted)
- IMPORTANT: Comments on appearance are stereotypes when contextually
irrelevant

5. SEXUAL STEREOTYPES
- Women: passive, pure, judged for sexual freedom
- Men: active, "hunters", naturally sexual
- Includes sexual insults like "puttana", "zoccola", "figlio/a di puttana"

6. RELATIONAL STEREOTYPES
- Women: caring, faithful, need protection, submissive
- Men: dominant, autonomous, emotionally detached
- Includes judgments about singlehood, marriage motivations, legal bias
claims

ANNOTATION RULE: If multiple stereotypes appear, choose the one that appears
FIRST in the text.

IMPORTANT: End your response with ONLY the JSON output.
Format: JSON_OUTPUT: {"gs_value": X.XX, "gs_category": "category"}

user: |
Text to analyze: "{text}"

Following the guidelines above:
1. Does this text contain a gender stereotype?
2. How obvious/strong is it? (Consider: would most people recognize it?)
3. Which category does it belong to?
4. What GS value should it receive? (0.0 = none, 1.0 = maximum)

Provide your analysis ending with: JSON_OUTPUT: {"gs_value": <0.00-1.00>,
"gs_category": "<role|personality|competence|physical|sexual|relational|no>"}
user_with_context: |
Context: "{context}"

Text to analyze: "{text}"

Following the guidelines above and considering the context:
1. Does the main text contain a gender stereotype?
2. How does the context affect the interpretation?
3. How obvious/strong is the stereotype? (Consider: would most people recognize
it?)
4. Which category does it belong to?
5. What GS value should it receive? (0.0 = none, 1.0 = maximum)

Provide your analysis ending with: JSON_OUTPUT: {"gs_value": <0.00-1.00>,
"gs_category": "<role|personality|competence|physical|sexual|relational|no>"}

```

(c) Zero-Shot Prompt 3 (CONTINUOUS): Guidelines

```

system: |
You are an expert linguist specializing in gender stereotype detection in
Italian text.

After your reasoning, you MUST end with ONLY the JSON output on a new line.
Format: JSON_OUTPUT: {"gs_value": X.XX, "gs_category": "category"}

user: |
Analyze the following Italian text for gender stereotypes using step-by-step
reasoning:

Text: "{text}"

Follow these steps:
1. Identify if any gender-based generalizations are present
2. Determine if these generalizations are stereotypical
3. Assess the strength/obviousness of the stereotype
4. Classify the type of stereotype
5. Assign a numerical score

Categories:
- role: What men/women should do/be (family roles, social expectations)
- personality: Character traits by gender (emotional, rational, etc.)
- competence: Ability judgments (STEM, leadership, sports)
- physical: Appearance expectations (beauty, grooming, body)
- sexual: Sexuality and sexual behavior norms
- relational: Behavior in relationships and interactions
- no: No stereotype present

GS Value (0.0-1.0):
- 1.0 = Explicit, clear stereotype
- 0.75 = Strong but may have slight ambiguity
- 0.50 = Moderate, some would debate
- 0.25 = Weak, subtle, or context-dependent
- 0.0 = No stereotype

After your reasoning, provide ONLY: JSON_OUTPUT: {"gs_value": <0.0-1.0>,
"gs_category": "<category>"}

user_with_context: |
Analyze the following Italian text for gender stereotypes using step-by-step
reasoning. Consider the context provided.

Context: "{context}"

Text to evaluate: "{text}"

Follow these steps:
1. Consider how the context frames the main text
2. Identify if any gender-based generalizations are present in the main text
3. Determine if these generalizations are stereotypical
4. Assess the strength/obviousness of the stereotype
5. Classify the type of stereotype
6. Assign a numerical score

```

(b) Zero-Shot Prompt 2 (CONTINUOUS): CoT

```

system: |
Sei un esperto di stereotipi di genere nella cultura e lingua italiana.

Il tuo compito è analizzare testi brevi e identificare:
1. Il livello di stereotipo di genere (valore GS da 0.0 a 1.0)
2. La categoria dello stereotipo

CATEGORIE:
- role: Ruoli familiari e sociali (es. uomo capofamiglia, donna casalinga)
- personality: Tratti emotivi/comportamentali (es. donna emotiva, uomo
razionale)
- competence: Capacità basate sul genere (es. uomini bravi in matematica)
- physical: Aspetto fisico e cura personale (es. donne devono essere belle)
- sexual: Comportamento sessuale (es. donna pura, uomo cacciatore)
- relational: Dinamiche relazionali (es. uomo dominante, donna sottomessa)
- no: Nessuno stereotipo presente

VALORI GS:
- 1.0 = Stereotipo molto chiaro ed esplicito
- 0.75 = Stereotipo forte
- 0.50 = Stereotipo moderato, può essere dibattuto
- 0.25 = Stereotipo debole o sottile
- 0.0 = Nessuno stereotipo

IMPORTANTE: Termina la risposta SOLO con l'output JSON.
Formato: JSON_OUTPUT: {"gs_value": X.XX, "gs_category": "category"}

user: |
Analizza questo testo italiano:

"{text}"

Termina con: JSON_OUTPUT: {"gs_value": <0.0-1.0>, "gs_category":
"<role|personality|competence|physical|sexual|relational|no>"}
user_with_context: |
Analizza questo testo italiano considerando il contesto fornito.

Contesto: "{context}"

Testo da valutare: "{text}"

IMPORTANTE: Valuta il testo principale considerando il contesto fornito. Il
contesto aiuta a comprendere la situazione ma l'analisi deve concentrarsi
sul testo principale.

Termina con: JSON_OUTPUT: {"gs_value": <0.0-1.0>, "gs_category":
"<role|personality|competence|physical|sexual|relational|no>"}

```

(d) Zero-Shot Prompt 4 (CONTINUOUS): Italian

Figure 2: Zero-Shot CONTINUOUS Classification Prompts.

```

system: |
You are simulating a panel of 4 annotators evaluating gender stereotypes in
Italian text.

Some texts are clearly stereotypical (all would agree), others are ambiguous
(annotators may disagree).
Your task is to estimate how many of 4 annotators would identify a gender
stereotype.

Categories (choose ONE):
- role: Social roles men/women should have
- personality: Character/emotional traits by gender
- competence: Ability judgments based on gender
- physical: Appearance/body expectations
- sexual: Sexual behavior and attitudes
- relational: Relationship dynamics and expectations
- no: No stereotype present

IMPORTANT: End your response with ONLY the JSON output.
Format: JSON_OUTPUT: {"gs_value": X.XX, "gs_category": "category"}
user: |
Text: "{text}"

Think like 4 different annotators:
- Annotator 1: Very sensitive to subtle stereotypes
- Annotator 2: Moderately sensitive, needs clearer signals
- Annotator 3: Moderately sensitive, context matters
- Annotator 4: Only identifies explicit stereotypes

For each annotator, would they say "yes, this is a stereotype" or "no"?

Count the "yes" responses:
- 4 yes   gs_value = 1.0
- 3 yes   gs_value = 0.75
- 2 yes   gs_value = 0.50
- 1 yes   gs_value = 0.25
- 0 yes   gs_value = 0.0

Then classify the category (if stereotype present).

End with: JSON_OUTPUT: {"gs_value": <0.0-1.0>, "gs_category": "<category>"}
user_with_context: |
Context: "{context}"

Text to evaluate: "{text}"

Consider the context when evaluating. Think like 4 different annotators:
- Annotator 1: Very sensitive to subtle stereotypes
- Annotator 2: Moderately sensitive, needs clearer signals
- Annotator 3: Moderately sensitive, context matters
- Annotator 4: Only identifies explicit stereotypes

For each annotator, considering the context, would they say "yes, this is a
stereotype" or "no"?

Count the "yes" responses:
- 4 yes   gs_value = 1.0
- 3 yes   gs_value = 0.75
- 2 yes   gs_value = 0.50
- 1 yes   gs_value = 0.25
- 0 yes   gs_value = 0.0

Then classify the category (if stereotype present).

End with: JSON_OUTPUT: {"gs_value": <0.0-1.0>, "gs_category": "<category>"}

```

(e) Zero-Shot Prompt 5 (CONTINUOUS): Perspectivist

```

system: |
You are an expert in detecting gender stereotypes in Italian text.
Your task is to analyze texts and provide two outputs:
1. A GS value (0.0 to 1.0) indicating the degree of gender stereotype present
2. The category of stereotype if present

GS Value Scale:
- 0.0 = No stereotype
- 0.25 = Weak stereotype (1 of 4 annotators would identify it)
- 0.50 = Moderate stereotype (2 of 4 annotators would identify it)
- 0.75 = Strong stereotype (3 of 4 annotators would identify it)
- 1.0 = Very strong stereotype (all annotators would identify it)

Categories:
- role: Social/cultural expectations about what men/women should do
- personality: Emotional/behavioral traits assigned by gender
- competence: Judgments about abilities based on gender
- physical: Expectations about appearance and body
- sexual: Attitudes about sexuality and sexual behavior
- relational: How genders should behave in relationships
- no: No stereotype present

Here are some examples:
{examples}

IMPORTANT: Respond ONLY with the JSON output. Do not include any explanations
before or after.
Format your response EXACTLY as: JSON_OUTPUT: {"gs_value": X.XX,
"gs_category": "category"}
user: |
Now analyze this Italian text for gender stereotypes:

Text: "{text}"

Provide your analysis as JSON_OUTPUT: {"gs_value": <number between 0.0 and
1.0>, "gs_category":
"<role|personality|competence|physical|sexual|relational|no>"}

```

(f) Few-Shot Prompt 1 (CONTINUOUS): Direct

Figure 2: Additional zero-shot CONTINUOUS Classification Prompts, and an example of few-shot (f).

```

system: |
You are an expert in detecting gender stereotypes in Italian.

TASK: Determine if the text contains a gender stereotype (0 = no, 1 = yes)

STEREOTYPE = 1 when the text makes claims about "men" or "women" in general
with stereotypical patterns.
- Are like: "Gli uomini sono razionali"
- Should do: "Le donne devono occuparsi della casa"
- Cannot do: "Un uomo non può essere sensibile"
STEREOTYPE = 0 when the text describes:
- Specific individuals: "Una donna entra nel negozio"
- Defined subgroups: "Gli uomini della mia famiglia"
- Gender mention without generalization: "Il ragazzo serve al tavolo"

CATEGORIES (select only the FIRST category that appears in the text):
- role: Family/social roles and expectations
- personality: Character traits (emotional, rational, etc.)
- competence: Abilities (STEM, leadership, sports, etc.)
- physical: Appearance expectations
- sexual: Sexuality and behavior norms
- relational: Relationship behaviors
- no: Use only when stereotype = 0

OUTPUT FORMAT:
{"stereotype": 0, "category": "no"}

Evaluate ONLY the given text. Do not infer missing context.

user: |
Analyze this Italian text for gender stereotype:

TEXT: "{text}"

JSON_OUTPUT: [{"stereotype": <0 or 1>, "category": "<category>"}]

user_with_context: |
Analyze this Italian text for gender stereotype within the provided context.

CONTEXT: "{context}"

TEXT: "{text}"

Evaluate ONLY the main text. Context is for interpretation, not
classification.

JSON_OUTPUT: [{"stereotype": <0 or 1>, "category": "<category>"}]

```

(a) Zero-Shot Prompt 1 (BINARY): Direct

```

system: |
You are an expert in detecting gender stereotypes in Italian.

TASK: Determine if the text contains a gender stereotype (0 = no, 1 = yes)

STEREOTYPE = 1 when the text makes claims about "men" or "women" in general.
- Are like: "Gli uomini sono razionali"
- Should do: "Le donne devono occuparsi della casa"
- Cannot do: "Un uomo non può essere sensibile"
STEREOTYPE = 0 when the text describes:
- Specific individuals: "Una donna entra nel negozio"
- Defined subgroups: "Gli uomini della mia famiglia"
- Gender mention without generalization: "Il ragazzo serve al tavolo"

CATEGORIES (select only the FIRST category that appears in the text):

1. ROLE STEREOTYPES
- Men: breadwinners, heads of household, financial support, protection
- Women: mothers, housewives, child-rearing, domestic work
- Note: Work aspects usually fall under "competence"

2. PERSONALITY STEREOTYPES
- Women: empathetic, sensitive, emotional, indecisive, caring
- Men: strong, assertive, rational, confident, unemotional
- Includes gendered insults (e.g., "femminuccia", "avere le palle")

3. COMPETENCE STEREOTYPES
- Women: suited for humanities, supportive roles
- Men: skilled in STEM, logic, technology, leadership
- Includes sports, games, professional abilities

4. PHYSICAL STEREOTYPES
- Women: slim, young, attractive, well-groomed (judged harshly)
- Men: robust, can be scruffy (more accepted)
- IMPORTANT: Comments on appearance are stereotypes when contextually
irrelevant

5. SEXUAL STEREOTYPES
- Women: passive, pure, judged for sexual freedom
- Men: active, "hunters", naturally sexual
- Includes sexual insults like "puttana", "zoccola", "figlio/a di puttana"

6. RELATIONAL STEREOTYPES
- Women: caring, faithful, need protection, submissive
- Men: dominant, autonomous, emotionally detached
- Includes judgments about singlehood, marriage motivations, legal bias
claims

OUTPUT FORMAT:
{"stereotype": 0, "category": "no"}

Evaluate ONLY the given text. Do not infer missing context.

user: |
Analyze this Italian text for gender stereotype:

TEXT: "{text}"

JSON_OUTPUT: [{"stereotype": <0 or 1>, "category": "<category>"}]

user_with_context: |
Analyze this Italian text for gender stereotype within the provided context.

CONTEXT: "{context}"

TEXT: "{text}"

Evaluate ONLY the main text. Context is for interpretation, not classification.

JSON_OUTPUT: [{"stereotype": <0 or 1>, "category": "<category>"}]

```

(c) Zero-Shot Prompt 3 (BINARY): Guidelines

```

system: |
You are an expert in detecting gender stereotypes in Italian.

After your reasoning, you MUST end with ONLY the JSON output on a new line.
Format: JSON_OUTPUT: {"stereotype": 0, "category": "no"}

user: |
Analyze the following Italian text for gender stereotypes using step-by-step
reasoning:

TEXT: "{text}"

Follow these steps:
1. Identify if any gender-based generalizations are present
2. Determine if these generalizations are stereotypical
3. Classify the type of stereotype if present
4. Make a binary decision (stereotype present or not)

Binary Classification:
- stereotype = 1: Gender stereotype IS present
- stereotype = 0: Gender stereotype is NOT present

Categories:
- role: What men/women should do/be (family roles, social expectations)
- personality: Character traits by gender (emotional, rational, etc.)
- competence: Ability judgments (STEM, leadership, sports)
- physical: Appearance expectations (beauty, grooming, body)
- sexual: Sexuality and sexual behavior norms
- relational: Behavior in relationships and interactions
- no: No stereotype present

After your reasoning, provide ONLY: JSON_OUTPUT: [{"stereotype": <0 or 1>,
"category": "<category>"}]

user_with_context: |
Analyze the following Italian text for gender stereotypes using step-by-step
reasoning. Consider the context provided.

CONTEXT: "{context}"

TEXT: "{text}"

Follow these steps:
1. Consider how the context frames the main text
2. Identify if any gender-based generalizations are present in the main text
3. Determine if these generalizations are stereotypical
4. Classify the type of stereotype if present
5. Make a binary decision (stereotype present or not)

Binary Classification:
- stereotype = 1: Gender stereotype IS present
- stereotype = 0: Gender stereotype is NOT present

Categories:
- role: What men/women should do/be (family roles, social expectations)
- personality: Character traits by gender (emotional, rational, etc.)
- competence: Ability judgments (STEM, leadership, sports)
- physical: Appearance expectations (beauty, grooming, body)
- sexual: Sexuality and sexual behavior norms
- relational: Behavior in relationships and interactions
- no: No stereotype present

After your reasoning, provide ONLY: JSON_OUTPUT: [{"stereotype": <0 or 1>,
"category": "<category>"}]

```

(b) Zero-Shot Prompt 2 (BINARY): CoT

```

system: |
Sei un esperto nel rilevamento di stereotipi di genere in italiano.

COMPITO: Determina se il testo contiene uno stereotipo di genere (0 = no, 1 =
s'i)

STEREOTIPO = 1 quando il testo fa affermazioni su "uomini" o "donne" in
generale.
- Sono come: "Gli uomini sono razionali"
- Dovrebbero fare: "Le donne devono occuparsi della casa"
- Non possono fare: "Un uomo non può essere sensibile"
STEREOTIPO = 0 quando il testo descrive:
- Individui specifici: "Una donna entra nel negozio"
- Sottogruppi definiti: "Gli uomini della mia famiglia"
- Menzione del genere senza generalizzazione: "Il ragazzo serve al tavolo"

CATEGORIE (seleziona solo la PRIMA categoria che compare nel testo):
- role: Ruoli e aspettative familiari/sociali
- personality: Tratti caratteriali (emotivo, razionale, ecc.)
- competence: Abilit'a (STEM, leadership, sport, ecc.)
- physical: Aspettative sull'aspetto fisico
- sexual: Sessualit'a e norme comportamentali
- relational: Comportamenti nelle relazioni
- no: Usa solo quando stereotype = 0

FORMATO OUTPUT:
{"stereotype": 0, "category": "no"}

Valuta SOLO il testo fornito. Non inferire contesto mancante.

user: |
Analizza questo testo italiano per rilevare stereotipi di genere:

TESTO: "{text}"

JSON_OUTPUT: [{"stereotype": <0 o 1>, "category": "<categoria>"}]

user_with_context: |
Analizza questo testo italiano per rilevare stereotipi di genere nel contesto
fornito.

CONTESTO: "{context}"

TESTO: "{text}"

Valuta SOLO il testo principale. Il contesto serve per l'interpretazione, non
per la classificazione.

JSON_OUTPUT: [{"stereotype": <0 o 1>, "category": "<categoria>"}]

```

(d) Zero-Shot Prompt 4 (BINARY): Italian

Figure 3: Zero-Shot BINARY Classification Prompts.

Table 10

Pairwise agreement of the Main Task (our annotations).

Ann_A	Ann_B	N	Cohen's κ
*A1	*A2	200	0.4505
*A1	*A3	200	0.5697
*A1	*A4	200	0.6139
*A2	*A3	200	0.5635
*A2	*A4	200	0.3891
*A3	*A4	200	0.4901

Table 11

Pairwise agreement of the Subtask (our annotations).

Ann_A	Ann_B	N	Cohen's κ
*A1	*A2	80	0.555
*A1	*A3	96	0.594
*A1	*A4	114	0.495
*A2	*A3	79	0.479
*A2	*A4	84	0.390
*A3	*A4	100	0.564

C. Additional Human Annotation Experiment

In this section, we describe an additional annotation experiment. By following the guidelines provided by the organizers, four Italian native speakers (co-authors of this paper) have annotated the 200 texts of the dev set. This work has been conducted to explore the subjective nature of the social phenomena the task is trying to model, i.e., identifying if a text contains a gender stereotype and of which kind.

To quantify the agreement between annotators, we compute Cohen's κ . Table 10 reports the pairwise agreement for the main task, i.e., has_stereotype: YES/NO (binary), for the total of 200 instances, while Table 11 reports the agreement for the subtask, i.e., the stereotype categories (multiclass), for 68 texts.⁵ Annotators are indicated with * to denote they are not the official annotators, but members of our team.

The agreement scores are consistently higher than 0.45, indicating a moderate agreement: the highest score is reached for the main task (0.61). We point out one notable exception: in both tasks, the same pair of annotators register the lowest agreement (0.39). In order to better understand what caused such misalignment, a collaborative approach was taken during a group meeting to share general insights, ensuring that any disagreements were addressed through discussions and ultimately resolved through consensus. In the following, we report considerations that emerged from this collective manual review.

We first highlight a clear tendency: one annotator assumed an over-identification position identifying an over-presence of stereotypes; while the other adopted the opposite, under-estimating their presence.

Example 1

IT: *Madonna che str0nza acida questa [USER] di questo profilo. Ma che cazzo ci campi a fare?*

EN: *Holy cow, what a sour bitch this [USER] on this profile is. What the hell are you doing here?*

This frequent pattern and the **great subjectivity of the task** itself, prompted us to question what we are truly modeling, and, in relation to the real-world application, if we are interested in limiting false negatives or maximizing true positives.

We acknowledge the strong subjective and ideological nature of the task, which could lead to marking as stereotype texts that express a **dissonant belief or faulty stance** with respect to the annotator's perspective, or, similarly, to not properly recognize a stereotype if the annotator agreed with the position expressed. We recognize the crucial role of the context to support the annotation. Indeed, in several instances, the content was labeled as stereotypical strictly in **relation to the context** (which was sometimes provided and often times was not).

Example 2

IT: *[Commento a post dalla dicitura "Quale sarebbe la tua reazione se adesso scoprissi di essere incinta?"] Vorrei tantissimo una femminuccia xké ho tre bellissimi maschietti ma ho paura,paura del 4 maschio,ho paura del giudizio della famiglia...insomma non sarei felicissima*

EN: *[Comment on a post with the caption "What would be your reaction if you found out you were pregnant now?"] I would really like a girl because I have three beautiful boys but I'm scared, scared of the fourth boy, scared of the family's judgement...in short, I wouldn't be very happy*

⁵68 were the common instances to compare where all annotators have labeled a category within the subtask.

In the absence of the preceding conversational or descriptive context, identifying a text as stereotypical becomes a very challenging (and arbitrary) task. In several cases, **utterances that were trying to counter a stereotype were expressing stereotypes themselves**: this particular but frequent scenario caused a different interpretation of the guidelines.

Example 3

IT: *Anche le figlie se si sposano e vanno in altra città, quando hanno i figli non le senti e non le vedi, e questione di carattere, donne ho uomini e lo stesso.*

EN: *Even daughters, if they get married and go to another city, when they have children you don't hear from them or see them, it's a question of character, women and men are the same.*

Moreover, the **use of irony and/or sarcasm** complicated stereotype identification, as stereotypical content may be expressed implicitly while simultaneously being challenged or mocked.

Example 4

IT: *[Commento ad articolo con titolo "Non esistono due cervelli, maschile e femminile. Il cervello è plastico e plasmato dall'ambiente".] Al Festival della Letteratura di Mantova la neuroscienziata Martina Ardizzi, docente all'Università di Padova, ha spiegato che differenze considerate biologiche, come emozioni, abilità spaziali o matematiche, dipendono in realtà dalla socializzazione. ...però il cervello, quando è nella testa di una donna, diventa miracolosamente... migliore!*

EN: *[Comment on an article titled "There are not two brains, male and female. The brain is plastic and shaped by the environment."] At the Literature Festival in Mantua, neuroscientist Martina Ardizzi, a professor at the University of Padua, explained that differences considered biological, such as emotions, spatial abilities, or mathematical abilities, are actually the result of socialization. ...but the brain, when it's in a woman's head, miraculously becomes... better!*

Regarding the categories proposed by the guidelines, we noticed that (1) occasionally the available categories did not cover the angle of the stereotype expressed, and (2) the difficulty of assigning only one category when overlapping dimensions were present. These two factors could potentially lead to inconsistent annotations, inconsistent data, and ultimately inconsistent models. Finally, in multiple samples, the stereotype identification was rooted in the literal reading of the text, but rather it was implicit: recognizing it involved several logical inferences. On this aspect, it would be an added value for the data to ask annotators to (1) highlight the span of text expressing the stereotype (i.e., a rationale), and (2) to write down the **implicit belief underlying the text**, as performed by Sap et al. [33].

Example 5

IT: *Ma trovo vergognoso strumentalizzare questa storia per farne l'ennesimo dogma ideologico per giustificare la violenza con cui le neo femministe rivendicano determinati torti. Accusando TUTTI i maschi indistintamente...*

EN: *But I find it shameful to exploit this story to make it yet another ideological dogma to justify the violence with which neo-feminists claim certain wrongs. Accusing ALL men without distinction...*

Overall, these observations confirm that identifying and categorizing stereotypes is a highly challenging task even for humans. Consequently, expecting models to learn this phenomenon reliably from imperfect and subjective annotations represents a sensitive direction.

We release our additional annotations publicly to support future research. For example, these annotations could be used to estimate a more robust GS Value by aggregating binary gender-stereotype labels from eight annotators instead of four, thereby better capturing the diversity and subjectivity of the task. Additionally, agreement on the development set between the original annotators (Table 8) was slightly higher than the agreement among the new annotators (Table 10). It would be interesting to investigate whether this difference affects model behavior and performance, and if so, to what extent.⁶

⁶The additional annotations are available at: <https://github.com/grecosalvatore/StereoBusters-GSI-Detect-Evalita2026>.

Team Prisma at GSI:detect: Comparing PB&J Persona-Based and Few-Shot Approaches to Gender Stereotype Detection at EVALITA 2026

Claudia Zaghi¹

¹*Independent Researcher*

Abstract

We present a comparative study of two approaches for automated gender stereotype detection in Italian text: persona-based modeling that simulates human annotator perspectives through psychological profiling, and few-shot learning using annotated examples. Our persona-based approach employs the Persona-based Judgment (PB&J) framework, which integrates established psychological constructs including Big Five personality traits, Schwartz values, and Primal World Beliefs to create synthetic annotator profiles. We submitted six runs: one aggregating four personas, four individual persona runs, and one few-shot baseline. Personas exhibited distinct detection behaviors (detection rates: 17–29%) but all under-detected stereotypes relative to the gold standard (70.1% stereotypical content). The few-shot approach outperformed persona runs on both the main task (intensity regression: NMSE 0.7041 vs. 2.04–2.24) and subtask (category classification: F1 Macro 0.6115 vs. 0.17–0.29). Persona modeling captures annotator variation but requires example-based calibration to match task conventions.

Keywords

gender stereotype detection, LLM personas, PB&J framework, few-shot learning, Italian NLP

1. Introduction

Gender stereotypes have shaped social discourse for centuries [1]. Social media has amplified stereotypical content [2], making automated detection both a social imperative and a computational challenge [3].

The Natural Language Processing (NLP) community has responded with corpora for probing bias in word embeddings [4], datasets for sexism detection [5], and evaluation frameworks for measuring stereotypical associations in language models [6]. Shared evaluation campaigns have emerged as key venues, including the NLPCC 2025 Gender Bias Mitigation Challenge for Chinese [7] and the GSI:detect task [8] at EVALITA 2026 [9] for Italian.

Gender stereotype detection is inherently subjective: legitimate disagreement among annotators reflects differences in interpretation, background, and sensitivity [10, 11]. Annotator characteristics significantly influence detection outcomes [12, 7], and GSI:detect embraces this by preserving individual judgments rather than collapsing them through majority voting.

We participated in GSI:detect with two approaches, motivated by the data harmonization literature emphasizing that consistency between sources is essential for valid predictions [13]:

1. **Persona-based zero-shot:** Constructing LLM personas using Persona-based Judgment (PB&J) framework [14] to match the demographic profile of the original annotators (three female, one male; ages 20–47) as much as possible.
2. **Few-shot learning:** Calibrating predictions using 35 stratified examples from the development data to align with annotation conventions.

Code and outputs are available at https://github.com/ClaZaghi/gsi_detect_evalita2026_prisma.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ cla.zaghi@gmail.com (C. Zaghi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Task and Data

The GSI:detect task focuses on gender stereotype detection in Italian short texts, adopting a perspectivist approach that preserves annotator disagreement rather than collapsing it through majority voting [15]. The task is articulated in two subtasks:

Main task (GS Detection): A regression task requiring systems to predict a GS value in $[0,1]$ quantifying the degree of stereotypical content. Values are computed from four independent binary annotations: complete agreement yields 0 or 1, while disagreement produces intermediate values (e.g., 0.25, 0.50, 0.75).

Subtask (GS Classification): A multi-class classification task requiring systems to assign texts to one of six stereotype categories: *role*, *personality*, *competence*, *physical*, *sexual*, and *relational*.

2.1. Dataset

The organisers provided approximately 1,000 Italian short texts collected from social media and informative websites, including both formal and informal language. The dataset is split as 20% development data and 80% test data. Each instance includes the text, individual annotations from four annotators, the aggregated GS value, the GS category, and a context flag indicating whether additional contextual information is provided. All four annotators are Italian native speakers, cisgender, and sensitive to gender-related issues; their demographic profile (three female, one male; ages 20–47; Master’s or PhD level) is detailed in Section 3.1.

2.2. Additional Resources

We adopted a closed-task setting, using no external training data or additional resources beyond the provided development set. Our few-shot system (Run 6) used 35 stratified examples from the development data, while persona-based systems (Runs 1–5) operated in a zero-shot setting.

All prompts were written in Italian and incorporated the annotation guidelines provided by the task organizers (see Appendix C for complete prompts).

3. System Description

3.1. Persona Construction with PB&J Framework

Our primary design goal was matching synthetic LLM personas to GSI:detect annotator demographics: four Italian native speakers (three female, one male; ages 20–40+; Master’s or PhD level) [8]. Since annotator identities were anonymized, demographic matching served as a proxy for preserving annotator diversity [13].

We linked demographic attributes to expected behavioral patterns (e.g., older annotators with more conservative thresholds) and applied the PB&J [14] to structure personas around three psychological dimensions:

- **Big Five Personality Traits** [16]: Conservative annotators received medium openness and high conscientiousness; sensitive annotators received very high openness with medium-high conscientiousness.
- **Schwartz Values** [17]: Conservative annotators emphasized tradition, security, and conformity; sensitive annotators prioritized self-direction and universalism.
- **Primal World Beliefs** [18]: Conservative annotators held higher “safe world” beliefs (higher evidence thresholds); sensitive annotators held lower beliefs (increased vigilance toward implicit bias).

Each persona included an Italian-language annotation rationale and experiential narrative grounded in these psychological dimensions. Complete persona definitions are provided in Appendix B.

Table 1 summarizes the four personas.

Table 1

PB&J persona profiles for Runs 1–5.

Attribute	A	B	C	D
Type	Balanced	Traditional	Sensitive	Academic
Gender	Female	Male	Female	Female
Age	35	47	26	29
Education	Master’s	Master’s	Master’s	PhD
Openness	Med-High	Medium	Very High	Very High
Conscientiousness	High	Very High	Med-High	High
Key Values	Universal.	Tradition	Self-dir.	Self-dir.
Safe World	Med-High	High	Med-Low	Low

3.2. Implementation

We submitted six runs using Claude 3.5 Sonnet via AWS Bedrock: five zero-shot persona-based runs and one few-shot baseline. All prompts were written in Italian and incorporated the annotation guidelines provided by task organizers in Italian (see Appendix A).

3.3. Run 1: Aggregated Personas (Zero-Shot)

All four personas defined in Appendix B annotated the test units independently, providing a binary judgment (stereotype present: yes/no). The final `gs_value` was computed as the proportion of personas detecting a stereotype, yielding values in {0.00, 0.25, 0.50, 0.75, 1.00}. Predictions were parallelized with 16 texts processed concurrently, each spawning 4 annotator threads.

3.4. Runs 2–5: Individual Personas (Zero-Shot)

Each run used a single persona predicting independently. For each text, the persona provided a binary stereotype judgment (yes/no), a category prediction, and an intensity estimate on a four-point scale for positive detections: 0.33 (slight/doubtful), 0.67 (moderate/clear), or 1.00 (strong/explicit). Negative detections received `gs_value` 0.00. Temperature parameters were calibrated to each persona’s expected consistency, with lower temperatures for more conservative annotators:

- **Run 2:** Persona A (Balanced) – Temperature 0.5
- **Run 3:** Persona B (Traditional) – Temperature 0.3
- **Run 4:** Persona C (Sensitive) – Temperature 0.7
- **Run 5:** Persona D (Academic) – Temperature 0.8

Complete persona definitions are provided in Appendix B. We discuss potential confounds from temperature variation and output scale choice in Section "Limitations".

3.5. Run 6: Few-Shot Baseline

The baseline used standard few-shot prompting with the translated annotation guidelines and 35 stratified examples, but without persona framing. Examples were drawn from the development set (5 per category, including “no stereotype”) using a fixed random seed, with order randomized for each prediction to reduce position bias. Temperature was set to 0.3. This run represents a closed-task setting with no external resources.

4. Results

4.1. Main Task: Stereotype Intensity Regression

The few-shot baseline outperformed all zero-shot persona runs on the regression task (NMSE 0.7041 vs. 2.04–2.24). Table 2 presents the main task results.

Table 2

Main task (regression) results on GSI:detect test set (810 samples).

Run	Approach	NMSE	MSE	Rank
Run 1	Aggregation (four-persona ensemble)	2.1152	0.3776	52/54
Run 2	Persona A	2.0382	0.3639	50/54
Run 3	Persona B	2.2409	0.4001	54/54
Run 4	Persona C	2.1134	0.3773	51/54
Run 5	Persona D	2.2322	0.3985	53/54
Run 6	Few-shot	0.7041	0.1257	20/54

Among zero-shot runs, results were mixed: Personas A and C slightly outperformed the aggregated four-persona ensemble (NMSE 2.0382 and 2.1134 vs. 2.1152), while Personas B and D underperformed (2.2409 and 2.2322). All zero-shot NMSE values exceeded 1.0, indicating performance worse than a baseline predicting the mean.

4.2. Subtask: Stereotype Category Classification

The few-shot baseline dominated the classification subtask, achieving first place among the 12 participating teams (F1 Macro 0.6115). Table 3 presents classification results.

Table 3

Subtask (classification) results on GSI:detect test set.

Run	Approach	F1 Macro	Rank
Run 1	Aggregation (four-persona ensemble)	0.2896	39/47
Run 2	Persona A	0.2358	44/47
Run 3	Persona B	0.1778	45/47
Run 4	Persona C	0.2291	43/47
Run 5	Persona D	0.1711	46/47
Run 6	Few-shot	0.6115	1/47

The aggregated four-persona ensemble outperformed all single-persona runs on classification (F1 0.2896 vs. 0.1711–0.2358). Unlike the main regression task, where outputs from single personas proved more effective, classification benefits from pooling diverse viewpoints to disambiguate overlapping categories.

Category detection difficulty varied with linguistic explicitness. Table 4 shows per-category F1 for Run 6 (few-shot), showing that concrete categories (e.g., physical) were easier to detect than implicit ones (e.g., relational). Table 4 shows per-category F1 for the best-performing run (Run 6).

5. Discussion

5.1. Detection Rate Analysis

We define detection rate as the percentage of texts a system classifies as containing stereotypes (i.e., $gs_value > 0$). Table 5 shows that all persona runs, both single and aggregated, systematically under-

Table 4

Per-category F1 scores for Run 6 (Few-shot).

Category	F1	Precision	Recall
physical	0.8404	0.8061	0.8778
role	0.7580	0.7411	0.7757
competence	0.7232	0.7788	0.6750
sexual	0.7143	0.7353	0.6944
personality	0.6757	0.6579	0.6944
relational	0.5690	0.7333	0.4648

detected stereotypes relative to the test data. The few-shot approach achieved 82.6% detection, substantially closer to the test data (70.1% stereotypical content), demonstrating effective threshold calibration through in-context examples.

Table 5

Stereotype detection rates across runs.

Run	No Stereotype	Stereotype	Detection Rate
Run 5: Persona D	669	141	17.4%
Run 3: Persona B	645	165	20.4%
Run 2: Persona A	629	181	22.3%
Run 1: Aggregation (four-persona ensemble)	607	203	25.1%
Run 4: Persona C	575	235	29.0%
Gold Standard	242	568	70.1%
Run 6: Few-shot	141	669	82.6%

Despite poor absolute performance, personas demonstrated behavioral differentiation. Persona B (conservative, high safe-world beliefs) achieved 20.4% detection, while Persona C (sensitive, low safe-world beliefs) achieved the highest rate at 29.0%. Persona D’s low detection rate (17.4%) despite a sensitive profile may reflect multiple factors: (1) its high temperature setting (0.8) increased output variability; (2) sequential processing meant Persona D ran last, potentially encountering rate limiting; and (3) post-processing rules defaulted unparseable JSON responses to non-detection (`gs_value = 0`).

5.2. Limitations

Our study has several limitations that should be considered when interpreting results.

- **Output scale mismatch.** Individual persona runs used a four-point scale ($\{0, 0.33, 0.67, 1.0\}$) while gold standard values, derived from four binary annotations, can only take values in $\{0, 0.25, 0.50, 0.75, 1.0\}$. This mismatch introduced irreducible error. However, this effect was minor relative to under-detection errors.
- **Temperature confounds.** Varying temperature settings across personas (0.3–0.8) confound interpretation of behavioral differences. Observed variation in detection rates may reflect temperature effects rather than psychological profile differences. Future work should isolate these factors by testing personas at fixed temperatures.
- **Persona D anomaly.** Persona D achieved the lowest detection rate (17.4%) despite having the most sensitive psychological profile (low safe-world beliefs, very high openness). This unexpected result may stem from its high temperature setting (0.8) introducing excessive randomness, or from rate limiting during sequential API calls causing degraded responses. We cannot distinguish between these explanations with current data.
- **Persona validation.** Mappings from demographics to psychological profiles were based on design assumptions, not empirically validated against actual annotator behavior.

- **Single model and language.** All experiments used Claude 3.5 Sonnet; results may not generalize to other LLMs. Additionally, the PB&J framework and psychological instruments were developed in English-language contexts, potentially limiting effectiveness for Italian annotation conventions.

6. Conclusion

We presented Team Prisma’s systems for the GSI:detect task at EVALITA 2026, comparing psychologically-grounded personas constructed with the PB&J framework against a few-shot baseline.

The few-shot system substantially outperformed all persona-based runs, ranking 20th of 54 on regression (NMSE 0.7041) and first among the 12 participating teams on classification (F1 Macro 0.6115). Among zero-shot approaches, we observed task-dependent trade-offs: single personas achieved lower regression error through continuous intensity outputs, while aggregating multiple personas improved classification by pooling diverse perspectives.

All persona runs systematically under-detected stereotypes (17.4%–29.0% vs. 70.1% gold standard), confirming that demographic alignment alone cannot replace task-specific calibration. Nevertheless, the 11.6% point spread between personas validates the PB&J framework’s ability to induce distinct behavioral patterns aligned with psychological profiles.

Future work will explore hybrid approaches combining persona diversity with few-shot calibration, aiming to capture legitimate annotator disagreement while achieving competitive absolute performance.

Acknowledgments

The author thanks the GSI:detect task organizers for providing the dataset and evaluation infrastructure.

Declaration on Generative AI

During the preparation of this work, the author used Claude 3.5 Sonnet (Anthropic) via AWS Bedrock for:

- **Writing assistance:** Editing manuscript text
- **Code development:** Assistance with Python for debugging and implementation

The author reviewed and edited all content and takes full responsibility for the publication.

References

- [1] L. Arcuri, M. R. Cadinu, Gli Stereotipi. Dinamiche psicologiche e contesto delle relazioni sociali, Il Mulino, Bologna, 1998.
- [2] M. R. Costa-jussà, An analysis of gender bias studies in natural language processing, *Nature Machine Intelligence* 1 (2019) 495–496.
- [3] S. L. Blodgett, S. Barocas, H. Daumé III, H. Wallach, Language (technology) is power: A critical survey of “bias” in NLP, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, 2020, pp. 5454–5476. URL: <https://aclanthology.org/2020.acl-main.485/>. doi:10.18653/v1/2020.acl-main.485.
- [4] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, A. Kalai, Man is to computer programmer as woman is to homemaker? debiasing word embeddings, *CoRR abs/1607.06520* (2016). URL: <http://arxiv.org/abs/1607.06520>. arXiv:1607.06520.
- [5] P. Chiril, V. Moriceau, F. Benamara, A. Mari, G. Origgi, M. Coulomb-Gully, An annotated corpus for sexism detection in French tweets, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis

- (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 1397–1403. URL: <https://aclanthology.org/2020.lrec-1.175/>.
- [6] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, K.-W. Chang, Gender bias in coreference resolution: Evaluation and debiasing methods, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 15–20. URL: <https://aclanthology.org/N18-2003/>. doi:10.18653/v1/N18-2003.
 - [7] Y. Li, G. Zhang, H. Hong, Y. Wang, C. Lin, Overview of the nlpcc 2025 shared task: Gender bias mitigation challenge, in: X.-L. Mao, Z. Ren, M. Yang (Eds.), Natural Language Processing and Chinese Computing, Springer Nature Singapore, Singapore, 2026, pp. 453–464.
 - [8] G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini, Gsi: detect at evalita 2026: Overview of the task on detecting gender stereotypes in italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [9] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [10] L. Aroyo, C. Welty, Truth is a lie: Crowd truth and the seven myths of human annotation, AI Magazine 36 (2015) 15–24. URL: <https://doi.org/10.1609/aimag.v36i1.2564>. doi:10.1609/aimag.v36i1.2564.
 - [11] A. Uma, T. Fornaciari, A. Dumitrache, T. Miller, J. Chamberlain, B. Plank, E. Simpson, M. Poesio, Learning from disagreement: A survey, Journal of Artificial Intelligence Research 72 (2021) 1385–1470.
 - [12] Z. Waseem, Are you a racist or am I seeing things? Annotator influence on hate speech detection on Twitter, in: Proceedings of the First Workshop on NLP and Computational Social Science, Austin, Texas, 2016, pp. 138–142.
 - [13] M. Y. Guan, V. Gulshan, A. M. Dai, G. E. Hinton, Who said what: Modeling individual labelers improves classification, 2018. URL: <https://arxiv.org/abs/1703.08774>. arXiv:1703.08774.
 - [14] B. Joshi, X. Ren, S. Swayamdipta, R. Koncel-Kedziorski, T. Paek, Improving language model personas via rationalization with psychological scaffolds, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2025, Association for Computational Linguistics, Suzhou, China, 2025, pp. 21747–21770. URL: <https://aclanthology.org/2025.findings-emnlp.1187/>. doi:10.18653/v1/2025.findings-emnlp.1187.
 - [15] F. Cabitza, A. Campagner, V. Basile, Toward a perspectivist turn in ground truthing for predictive computing, Proceedings of the AAAI Conference on Artificial Intelligence 37 (2023) 6860–6868. URL: <http://dx.doi.org/10.1609/aaai.v37i6.25840>. doi:10.1609/aaai.v37i6.25840.
 - [16] L. R. Goldberg, The structure of phenotypic personality traits, American Psychologist 48 (1993) 26–34.
 - [17] S. H. Schwartz, Universals in the content and structure of values: Theoretical advances and empirical tests in 20 countries, Advances in Experimental Social Psychology 25 (1992) 1–65.
 - [18] J. D. Clifton, J. D. Baker, C. L. Park, et al., Primal world beliefs, Psychological Assessment 31 (2019) 82–99.

A. Annotation Guidelines

All prompts incorporated the following Italian annotation guidelines, adapted from the task organizers’ original guidelines. These were included verbatim in all persona-based and few-shot prompts.

Listing 1: Italian annotation guidelines included in all prompts

LINEE GUIDA - Categorie di Stereotipi

Gli stereotipi di genere sono credenze rigide e generalizzate riguardo ai ruoli, comportamenti, abilità o caratteristiche che uomini e donne "dovrebbero" avere in base al loro sesso o genere.

Ogni esempio deve essere assegnato a una sola categoria, anche se nello stesso testo compaiono più stereotipi.

Regola generale: se nello stesso esempio compaiono diverse categorie di stereotipi, gli annotatori devono selezionare la categoria che appare per prima nel testo.

Esempio:

"Le donne sono troppo emotive per essere buone leader, e gli uomini dovrebbero sempre mantenere la famiglia."

-> Contiene sia stereotipo di competenza che di ruolo, ma poiché lo stereotipo di competenza appare per primo, deve essere assegnato alla categoria "Competenza".

NB: Nel caso dei commenti, è importante considerare se il commento è pertinente al contesto.

- Se qualcuno commenta l'aspetto fisico di una donna in un contesto in cui è irrilevante (es. durante un discorso politico) -> stereotipo
- Se l'aspetto è contestualmente rilevante (es. commento sull'outfit in una sfilata di moda) -> non stereotipo

1. Stereotipi di Ruolo (role)

Aspettative sociali e culturali che definiscono quali ruoli familiari o sociali sono "appropriati" per uomini e donne.

- Uomini: capifamiglia, responsabili del sostegno economico, della protezione e delle decisioni familiari.
- Donne: madri, casalinghe, responsabili della cura dei figli e del lavoro domestico.

2. Stereotipi di Personalità (personality)

Attribuzione di tratti emotivi o caratteriali basati sul genere.

Include insulti: "femminuccia", "checca isterica", "frocio", "fighette" per uomini; "avere le palle", "donna con le palle" per donne.

- Donne: empatiche, sensibili, dolci, premurose, emotive, indecise.
- Uomini: forti, assertivi, autorevoli, razionali, poco emotivi.

3. Stereotipi di Competenza (competence)

Giudizi generalizzati sulle abilità professionali, intellettuali o fisiche basati sul genere. Include STEM, sport, scacchi.

- Donne: adatte alle materie umanistiche, ruoli di supporto.
- Uomini: naturalmente portati per scienza, logica, tecnologia, leadership.

4. Stereotipi Fisici (physical)

Aspettative relative al corpo, all'aspetto e alla cura di sé.

- Donne: magre, giovani, attraenti, curate. Giudicate più severamente.
- Uomini: robusti, trasandati, "accettabili così come sono."

5. Stereotipi Sessuali (sexual)

Stereotipi relativi al desiderio, sessualità e comportamento sessuale.

SEMPRE include: "puttana", "zoccola", "troia" (anche come sinonimo di "stronza"). Anche "figlio/a di puttana" rientra qui.

- Donne: passive, pure, giudicate per la libertà sessuale.
- Uomini: attivi, "cacciatori", naturalmente portati al sesso.

6. Stereotipi Relazionali (relational)

Modelli rigidi di comportamento nelle relazioni interpersonali e romantiche. Include: "zitelle" vs "scapoli", matrimoni per soldi, bias legali ("privilegi femminili").

- Donne: premurose, fedeli, bisognose di protezione, sottomesse.
- Uomini: dominanti, competitivi, autonomi, emotivamente distaccati.

B. Complete Persona Definitions

Table 1 in the main text summarizes persona demographics and psychological profiles. Below we provide the complete Italian-language definitions used in the prompts.

B.1. Persona A (Balanced)

Listing 2: Persona A complete definition

PROFILO PERSONALE
DATI DEMOGRAFICI:
- Genere: femmina
- Eta: 35 anni
- Formazione: laurea magistrale
- Provenienza: Nord Italia
- Background religioso: No
PERSONALITA (Big Five):
- openness: medio-alto
- conscientiousness: alto
- extraversion: medio
- agreeableness: medio-alto
- neuroticism: basso
VALORI (Schwartz):
- universalismo, benevolenza, sicurezza
PRIMAL WORLD BELIEFS:
- safe_world: medio-alto
- enticing_world: medio
- alive_world: medio
IL TUO APPROCCIO ALL'ANNOTAZIONE:
Applico una valutazione equilibrata degli stereotipi di genere, identificando quelli espliciti con coerenza. La mia formazione in scienze sociali e la mia esperienza professionale mi permettono di riconoscere gli stereotipi evidenti mantenendo un approccio critico ma non eccessivamente rigido. Valuto il contesto e l'intenzionalità del messaggio.
ESPERIENZE:
Lavoro nel settore dell'educazione e ho esperienza diretta con questioni di genere.

B.2. Persona B (Traditional)

Listing 3: Persona B complete definition

PROFILO PERSONALE
DATI DEMOGRAFICI:
- Genere: maschio
- Eta: 47 anni
- Formazione: laurea magistrale
- Provenienza: Centro Italia
- Background religioso: Si
PERSONALITA (Big Five):
- openness: medio
- conscientiousness: molto alto
- extraversion: medio-basso
- agreeableness: medio-alto
- neuroticism: basso
VALORI (Schwartz):
- tradizione, sicurezza, benevolenza, conformita
PRIMAL WORLD BELIEFS:

```
- safe_world: alto
- enticing_world: medio-basso
- alive_world: medio
```

IL TUO APPROCCIO ALL'ANNOTAZIONE:

Richiedo prove esplicite e chiare per identificare uno stereotipo di genere. La mia esperienza professionale e il mio approccio metodico mi portano a distinguere tra commenti infelici e veri stereotipi dannosi. Preferisco essere conservativo nelle mie valutazioni per evitare false accuse.

ESPERIENZE:

Sono cresciuto in un contesto tradizionale ma ho sempre cercato di essere aperto al cambiamento.

B.3. Persona C (Sensitive)

Listing 4: Persona C complete definition

PROFILO PERSONALE

DATI DEMOGRAFICI:

```
- Genere: femmina
- Eta: 26 anni
- Formazione: laurea magistrale
- Provenienza: Sud Italia
- Background religioso: No
```

PERSONALITA (Big Five):

```
- openness: molto alto
- conscientiousness: medio-alto
- extraversion: medio-alto
- agreeableness: medio
- neuroticism: medio
```

VALORI (Schwartz):

```
- autodirezione, universalismo, stimolazione, benevolenza
```

PRIMAL WORLD BELIEFS:

```
- safe_world: medio-basso
- enticing_world: alto
- alive_world: alto
```

IL TUO APPROCCIO ALL'ANNOTAZIONE:

Sono attenta agli stereotipi impliciti e alle sfumature linguistiche che rivelano pregiudizi di genere nascosti. La mia formazione teorica e la mia sensibilità mi permettono di identificare stereotipi anche quando non sono esplicitamente espressi.

ESPERIENZE:

Sono cresciuta in un contesto dove gli stereotipi di genere erano molto presenti. Ho studiato sociologia e gender studies.

B.4. Persona D (Academic)

Listing 5: Persona D complete definition

PROFILO PERSONALE

DATI DEMOGRAFICI:

```
- Genere: femmina
- Eta: 29 anni
- Formazione: dottorato di ricerca
- Provenienza: Nord Italia
- Background religioso: No
```

PERSONALITA (Big Five):

```
- openness: molto alto
- conscientiousness: alto
```

- extraversion: medio
- agreeableness: medio-basso
- neuroticism: medio-alto

VALORI (Schwartz):

- autodirezione, universalismo, stimolazione, realizzazione

PRIMAL WORLD BELIEFS:

- safe_world: basso
- enticing_world: molto alto
- alive_world: molto alto

IL TUO APPROCCIO ALL'ANNOTAZIONE:

La mia formazione accademica approfondita in studi di genere e linguistica critica mi permette di identificare anche gli stereotipi più sottili e insidiosi. Gli stereotipi benevoli sono comunque stereotipi. Riconosco le microaggressioni e le assunzioni implicite sui ruoli di genere.

ESPERIENZE:

Ho dedicato anni allo studio critico delle rappresentazioni di genere nel discorso pubblico italiano.

C. Prompt Templates

C.1. Persona Prompt Structure (Runs 1–5)

Listing 6: Persona prompt template structure

```
{persona.get_system_context()}

LINEE GUIDA PER L'ANNOTAZIONE

DEFINIZIONE:
Gli stereotipi di genere sono convinzioni rigide e generalizzate sui ruoli, comportamenti, abilità o caratteristiche che uomini e donne "dovrebbero" avere in base al sesso o genere.

REGOLE:
1. ASSEGNA UNA SOLA CATEGORIA per testo
2. Se multipli stereotipi -> scegli il PRIMO che appare
3. CONTESTO è critico per stereotipi fisici

CATEGORIE:
1. role - Stereotipi di RUOLO
2. personality - Stereotipi di PERSONALITÀ
3. competence - Stereotipi di COMPETENZA
4. physical - Stereotipi FISICI
5. sexual - Stereotipi SESSUALI
6. relational - Stereotipi RELAZIONALI

CONSULTA LE LINEE GUIDA:
{guidelines_it}

TESTO DA ANALIZZARE:
"{text}"

Analizza questo testo con la TUA prospettiva unica.

Rispondi SOLO in formato JSON:
{
  "contains_stereotype": "yes" | "no",
  "category": "role" | "personality" | "competence" |
    "physical" | "sexual" | "relational" | null,
  "confidence": "high" | "medium" | "low",
  "reasoning": "Breve spiegazione dalla tua prospettiva"
}
```


C.2. Few-Shot Prompt Structure (Run 6)

Listing 7: Few-shot prompt template structure

Sei un'esperta nell'identificazione di stereotipi di genere in testi italiani.

DEFINIZIONE

Gli stereotipi di genere sono convinzioni rigide e generalizzate sui ruoli, comportamenti, abilità o caratteristiche che uomini e donne "dovrebbero" avere in base al sesso o genere.

COMPITO

1. Assegna un valore `gs_value` da 0.00 a 1.00:

- 0.00 = nessuno stereotipo
- 0.25 = stereotipo lieve
- 0.50 = stereotipo moderato
- 0.75 = stereotipo forte
- 1.00 = stereotipo massimo

2. Classifica in UNA categoria:

- no: Nessuno stereotipo
- role: Ruoli familiari/sociali
- personality: Tratti emotivi/caratteriali
- competence: Abilità professionali/intellettuali
- physical: Aspetto/corpo/cura personale
- sexual: Sessualità/comportamento sessuale
- relational: Comportamento in relazioni

CONSULTA LE LINEE GUIDA:

{guidelines_it}

ESEMPI ANNOTATI (dal dataset GSI DEV)

ESEMPIO 1:

Testo: "{example_1_text}..."

-> `gs_value`: {example_1_value}

-> `gs_category`: {example_1_category}

[... 35 esempi totali, 5 per categoria ...]

TESTO DA ANALIZZARE

"{text}"

RISPOSTA

Analizza il testo seguendo gli esempi sopra.

Rispondi SOLO in formato JSON:

```
{
  "gs_value": "0.XX",
  "gs_category": "category_name",
  "reasoning": "Breve spiegazione (2-3 frasi)"
}
```

Tiz at GSI:detect: Modeling Gender Stereotype Detection as Multi-Category Gender Stereotype Scoring

Tiziano Labruna

University of Padua, 63 Via Trieste, Padova, 35121, Italy

Abstract

Gender stereotypes in language contribute to the reinforcement of social inequalities and represent a critical challenge for natural language processing systems. The GSI:DETECT shared task addresses this issue by evaluating systems on their ability to detect and classify gender stereotypes in short Italian texts, combining a regression-based main task with a multi-class classification subtask. In this paper, we present an LLM-based approach inspired by strategy-aware reasoning methods, which explicitly models different types of gender stereotypes through a multi-step prompting framework. Our method decomposes the detection process by independently analyzing multiple stereotype categories and aggregating their contributions to estimate the overall degree of gender stereotype. The proposed approach is evaluated on both the main task of gender stereotype detection and the subtask of stereotype classification, showing the potential of structured reasoning and category-aware analysis for addressing nuanced forms of stereotype in text.

Keywords

Gender Stereotype Detection, Gender Stereotypes, Large Language Models, Prompting Strategies, Strategy-aware Reasoning, Regression-based Stereotype Modeling

1. Introduction

Gender stereotypes encoded in language play a key role in shaping and reinforcing social norms, influencing how roles, abilities, and behaviors are associated with different genders [1, 2]. As language technologies are increasingly deployed in real-world applications, the presence of such stereotypes has raised growing concerns regarding the fairness and social impact of NLP systems [3, 4]. Consequently, the automatic detection of gender stereotype has emerged as an important research problem, attracting sustained attention from the NLP community in the form of surveys, dedicated datasets, benchmarks, and shared tasks [5, 6, 7].

Most existing approaches to gender stereotype detection frame the task as a binary or categorical classification problem, typically relying on majority-voted annotations that collapse annotator disagreement into a single hard label [8, 9]. However, recent work has highlighted how this practice may obscure the inherently subjective and gradient nature of social bias, leading to an oversimplified representation of the phenomenon [10]. In response, several studies have advocated for perspectivist annotation frameworks that explicitly preserve annotator disagreement and model bias as a continuum rather than a binary property [11, 12]. Within this line of work, the GSI:DETECT shared task [13] at the EVALITA 2026 workshop [14] adopts a regression-based formulation, requiring systems to predict a continuous score reflecting the degree of gender stereotypical content in a text, alongside an auxiliary classification task identifying the stereotype category.

In parallel, Large Language Models (LLMs) have recently shown strong capabilities in a great number of different complex reasoning tasks. Despite these advances, the reliability of LLM-based judgments remains highly sensitive to the prompting strategy, making it crucial to explicitly assess their trustworthiness when no carefully designed prompting or supervision is applied [15, 16, 17]. Prior work has demonstrated that prompting strategies, in-context learning, and chain-of-thought reasoning can substantially improve LLM performance on tasks requiring structured reasoning [18, 19]. Building on these

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ tiziano.labruna@unipd.it (T. Labruna)

🆔 0000-0001-7713-7679 (T. Labruna)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

advances, several approaches have explored the integration of LLM-based predictions with supervised models to improve robustness and alignment with human judgments in bias detection settings [20]. Furthermore, strategy-aware reasoning frameworks such as PCoT (Persuasion Chain-of-thought) [21] have shown that explicitly modeling high-level strategies can improve performance in related tasks like disinformation detection.

Inspired by these two lines of work, we propose a category-aware prompting approach tailored to gender stereotype detection. Unlike PCoT, our method does not consider persuasion strategies but instead focuses on the six gender stereotype categories defined in the GSI:DETECT task: *role*, *personality*, *competence*, *physical*, *sexual*, and *relational* stereotypes. Rather than injecting all categories simultaneously, we treat each category independently and perform a structured two-step prompting process for each of them.

In the first step, the LLM is prompted to generate an analytical paragraph assessing the presence of gender stereotype in the input text with respect to a single stereotype category. This step explicitly injects category-specific knowledge by including a description of the target stereotype type in the prompt, encouraging focused and interpretable reasoning. In the second step, the generated analysis is used to guide a decision or score generation, depending on the experimental setting.

For the zero-shot variant, the model outputs a binary decision indicating whether gender stereotype is present for the given category. These decisions are then aggregated across categories to determine whether the text contains gender stereotypes. For the fine-tuned variant, the model produces a numerical score on a fixed scale for each category. The resulting six-dimensional score vector is subsequently used as input to a supervised regression model, implemented as a multi-layer perceptron, which is trained on the development data to predict the final GS value in the range $[0, 1]$.

We also participate in the gender stereotype classification subtask by leveraging the category-wise scores produced by the fine-tuned model. The predicted stereotype category is selected as the one associated with the highest score, with ties resolved randomly. While this strategy is simple, it allows us to assess the extent to which category-specific stereotype signals captured during the main task can be reused for stereotype classification.

The main contributions of this work can be summarized as follows:

- We propose a category-aware, multi-step prompting framework for gender stereotype detection that explicitly models the six stereotype categories defined in the GSI:DETECT task.
- We introduce a two-stage LLM-based reasoning process in which category-specific analytical explanations are generated and subsequently used to guide stereotype detection and scoring.
- We present both a zero-shot and a fine-tuned variant of the proposed approach, showing how LLM-generated category-wise signals can be aggregated to model gender stereotype as a continuous phenomenon.
- We leverage category-specific stereotype scores to address the gender stereotype classification subtask, enabling reuse of the same signals across multiple evaluation settings.

2. Data and Resources

2.1. Dataset

We base our experiments on the dataset released for the GSI:DETECT shared task [13], which targets the detection and classification of gender stereotypes in short Italian texts. The dataset, considering only the portion made available to participants for development, consists of 200 annotated texts.

Each instance is provided in `.jsonl` format and includes the following fields: a unique identifier, the textual content, a list of four independent binary annotations indicating the presence or absence of a gender stereotype, a continuous GS value in the range $[0, 1]$ derived from the aggregation of annotators' judgments, a gender stereotype category label, and a binary flag specifying whether additional contextual information is required to interpret the text. In cases where context is required, it is explicitly included in the textual content and precedes the main text.

The GS value reflects the degree of stereotypical content and is computed by combining the four binary annotations rather than applying majority voting. Consequently, GS values close to the extremes (0 or 1) correspond to high inter-annotator agreement, while intermediate values capture varying degrees of disagreement. For the classification subtask, each instance is assigned to one of six predefined gender stereotype categories, i.e. *role*, *personality*, *competence*, *physical*, *sexual*, and *relational*, using majority voting or adjudication by a super-judge when necessary.

2.2. Data Augmentation

Given the limited size of the development data and the need to train a supervised regression model for the fine-tuned variant of our approach, we performed a form of data augmentation at the score level. Starting from the original 200 instances, we leveraged the category-wise scores produced by the LLM (which will be more precisely described in the next sections) to generate additional training samples while preserving the relative relationships among stereotype categories.

For each instance, we consider the six category-specific scores (one per gender stereotype category) and generate augmented variants by applying small bounded perturbations to these scores. Augmented samples are retained only if they preserve the original ranking relationships among categories, thereby maintaining consistency in the relative strength of different stereotype signals. This constraint prevents unrealistic inversions while allowing controlled variability in the score space.

This process results in multiple augmented versions for each original instance, effectively increasing the amount of training data available for the supervised learner without altering the underlying semantic content of the texts.

2.3. Data Splits

For the fine-tuning experiments, we construct train, development, and test splits at the level of original instances, ensuring that all augmented variants derived from the same base text are assigned to the same split. This prevents information leakage across splits and preserves a clean evaluation setup. We adopt a 60% / 20% / 20% split for training, development, and testing, respectively, using a fixed random seed for reproducibility. For each split, both the augmented score representations and the corresponding duplicated annotations are generated and aligned via consistent identifiers.

2.4. Computational Resources

All LLM-based experiments were conducted using the *Gemma-3.1-12B* model, accessed through the Hugging Face ecosystem without any task-specific fine-tuning. The model was employed exclusively in inference mode and used to generate both category-specific analyses and numerical stereotype scores through prompting. Inference was performed on a single NVIDIA A40 GPU, using mixed-precision computation to ensure efficiency and stability.

No external fine-tuning of the language model was performed. The only supervised component of our system is the multi-layer perceptron used in the fine-tuned variant to map category-wise LLM scores to a final GS value in the $[0, 1]$ range.

3. Methodology

Our approach models gender stereotype as a multidimensional phenomenon by decomposing the detection process across the six gender stereotype categories defined in the GSI:DETECT task: *role*, *personality*, *competence*, *physical*, *sexual*, and *relational*. Instead of prompting the model with all categories simultaneously, we adopt a category-aware, one-task-at-a-time strategy inspired by strategy-aware reasoning frameworks. Each category is analyzed independently through a structured two-step prompting procedure, and the resulting signals are subsequently aggregated to produce the final prediction.

3.1. Overview of the Framework

Given an input text, the system performs the following steps:

1. Independently analyze the text with respect to each gender stereotype category using a category-specific prompt.
2. Generate an explicit analytical explanation describing whether and how the stereotype category appears in the text.
3. Use the generated analysis to guide a second prompt that produces either a binary decision or a numerical score.
4. Aggregate category-level outputs to obtain a final gender stereotype score and category prediction.

This design encourages focused reasoning, reduces interference between categories, and makes intermediate model outputs interpretable.

3.2. Step 1: Category-Specific Analysis (Knowledge Infusion)

For each input text, we run six independent prompts, one for each gender stereotype category. Each prompt injects structured knowledge derived from the official task guidelines, explicitly defining the stereotype category, providing examples, and describing its social implications.

The system prompt establishes the model as an expert in a specific stereotype category, while the user prompt asks for a conservative assessment of whether that category is present in the text.

An example of a category-specific prompt for `[ROLE_STEREOTYPES]` is reported below:

System prompt:

You are an assistant who detects gender stereotypes in text. Gender stereotypes are rigid and generalized beliefs about the roles, behaviors, abilities, or characteristics that men and women “should” have based on their sex or gender.

Your expertise and focus is on `[ROLE_STEREOTYPES]`, meaning social and cultural expectations that define which family or social roles are “appropriate” for men and women.

Men: breadwinners, heads of the household.

Women: mothers, housewives, responsible for domestic work.

You are the expert who detects `[ROLE_STEREOTYPES]`.

User prompt:

Given a certain text, critically evaluate its gender stereotype potential. Additionally, analyze whether the text employs the `[ROLE_STEREOTYPES]` category. Explain how `[ROLE_STEREOTYPES]` appears or does not appear in the text. Be conservative in your assessment: if you are not fully certain that the category is used, assume it is not present.

Analogous prompts are used for the remaining five categories, with the placeholder `[CATEGORY]` replaced by the corresponding stereotype definition and examples. The output of this step is a short analytical paragraph describing the presence or absence of the target stereotype category.

3.3. Step 2: Decision or Scoring Based on the Analysis

The analysis generated in Step 1 is then passed to a second prompt, together with the original text. This second step uses a general gender stereotype detection instruction and leverages the previously generated analysis as explicit reasoning input. We consider two variants for this step.

Binary (Zero-Shot) Variant. In the zero-shot setup, the model is asked to produce a binary decision indicating whether gender stereotype is present with respect to the analyzed category.

System prompt: You are a Gender Stereotype Detector. Your goal is to detect if a text presents gender stereotypes. Gender stereotypes are rigid and generalized beliefs about the roles, behaviors, abilities, or characteristics that men and women "should" have based on their sex or gender. You will respond with either Yes or No.

User prompt: Given a text and an analysis of potential gender stereotypes or biased expressions it may contain, you must respond with ONLY Yes or No based on if you think that there is gender stereotype in the text or not.

The output is mapped to a binary value, where *Yes* corresponds to 1 and *No* to 0.

Score-Based Variant. In the score-based setup, used for the fine-tuned approach, the model produces a numerical score reflecting the degree of gender stereotype.

System prompt: You are a Gender Stereotype Detector. Your goal is to evaluate the degree of gender stereotypes present in a text. Gender stereotypes are rigid and generalized beliefs about the roles, behaviors, abilities, or characteristics that men and women should have based on their sex or gender. You will provide a score from 1 to 10.

User prompt: Given a text and an analysis of potential gender stereotypes or biased expressions it may contain, you must respond with a number from 1 to 10 based on the degree of gender stereotype in the text, preceded by a brief explanation of your reasoning. Give your answer in the form of a dictionary: {"explanation": "Brief explanation of the reasoning that led you to this score.", "response": "Score from 1 to 10."}

Each category thus produces either a binary indicator or a numerical score.

3.4. Zero-Shot Aggregation

In the binary setting, each text produces six binary outputs, one per category. We experiment with multiple aggregation strategies to derive a final prediction. The specific strategies will be presented in the next section. For category prediction, the stereotype category associated with the highest score is selected, with ties broken randomly.

3.5. Supervised Aggregation via Multi-Layer Perceptron

In the fine-tuned variant, the category-wise scores produced by the LLM are further combined using a supervised learning component. Specifically, we train a Multi-Layer Perceptron (MLP) to map the six-dimensional vector of category scores to the final gender stereotype signal required by the main task.

For each text, the MLP receives as input the six scores generated by the LLM, one for each gender stereotype category. In addition to these raw scores, we optionally augment the input with simple descriptive statistics computed over the six values. Specific experimental setting details can be found in the next section.

Model Architecture. The aggregation model is implemented as a feed-forward neural network composed of:

- an input layer matching the feature dimensionality,
- one or two hidden layers with ReLU activations,
- a final linear layer producing four outputs.

The model outputs correspond to the four binary annotations provided by the human annotators. This design choice allows the network to learn from annotator-level disagreement rather than directly regressing to a single aggregated label.

4. Experimental Setup

This section describes the experimental configurations adopted for both the zero-shot and the fine-tuned variants of our approach.

4.1. Zero-Shot Aggregation Strategies

In the zero-shot setting, the methodology described in Section 3 is applied without using any task-specific training data. For the main task, the LLM produces a binary score (0 or 1) for each of the six gender stereotype categories. As a result, each input text is associated with a six-dimensional binary vector.

Since the task requires a single continuous score in the $[0, 1]$ range, we explore multiple heuristic aggregation strategies to combine the six binary signals into a final prediction. The following five strategies are considered:

- **Max-One**: the final score is set to 0 if at most one category score is 1, and to 1 otherwise. (*run_1*)
- **Max-Two**: the final score is set to 0 if at most two category scores are 1, and to 1 otherwise. (*run_2*)
- **Max-Three**: the final score is set to 0 if at most three category scores are 1, and to 1 otherwise. (*run_3*)
- **All Zeros**: the final score is set to 0 if all six category scores are 0 (i.e., no category is flagged as biased), and to 1 otherwise. (*run_4*)
- **Average**: the final score is computed as the average of the six binary values, rounded to two decimal places. (*run_5*)

As this is a purely zero-shot approach, no development data is used to select or tune these strategies. Consequently, the chosen aggregation rules are not optimized on the task data, and other untested strategies might potentially yield better performance.

4.2. Fine-Tuned Aggregation Experiments

The fine-tuned variant represents the primary focus of this work. In this setting, the aggregation of category-wise signals is learned via a supervised Multi-Layer Perceptron, as described in Section 3.5.

Unlike the zero-shot setup, the MLP operates on the six continuous scores in the $[1, 10]$ range generated by the LLM for each category. We conduct an extensive set of experiments to identify the most effective configuration. Specifically:

- **add_scores** indicates that the six raw category scores are used as input features.
- **add_stats** indicates that three additional statistics computed over the six scores (mean, variance, entropy) are included.
- **threshold** denotes that the annotator-level probabilities predicted by the MLP are binarized using the specified threshold before aggregation.
- **augmented** indicates that the MLP is trained on the augmented version of the training and development sets, generated as described in Section 3.5.

For all experiments, hyperparameters are selected via grid search on the development set using the following configuration:

```
hidden_dim: {32, 64, 128}
hidden_layers: {1, 2}
lr: {1e-3, 1e-4}
batch_size: {64, 128}
```

Performance is evaluated using the $1/1+\text{MSE}$ (Mean Squared Error) metric, in line with the official evaluation metric of the task. Table 1 reports the scores obtained across all fine-tuning configurations.

Configuration	$1/(1 + \text{MSE})$
base + add_scores (<i>run_1</i>)	0.8137
base + add_stats (<i>run_2</i>)	0.8317
base + add_scores + add_stats (<i>run_3</i>)	0.8316
base + threshold 0.4	0.7407
base + threshold 0.5	0.7767
base + threshold 0.6	0.7767
base + threshold 0.7	0.7881
base + threshold 0.8	0.8000
base + threshold 0.9	0.7111
base + add_scores + threshold 0.5	0.7729
base + add_stats + threshold 0.5	0.7729
base + add_stats + threshold 0.8	0.7729
base + add_scores + add_stats + threshold 0.5	0.7767
augmented + add_scores	0.8215
augmented + add_stats	0.8169
augmented + add_scores + add_stats	0.8211
augmented + threshold 0.4	0.7148
augmented + threshold 0.5	0.7431
augmented + threshold 0.6	0.7645
augmented + threshold 0.7	0.7541
augmented + threshold 0.8	0.7556
augmented + add_scores + threshold 0.5	0.7568
augmented + add_stats + threshold 0.5	0.7622
augmented + add_stats + threshold 0.6	0.7741
augmented + add_scores + add_stats + threshold 0.5	0.7332

Table 1

Performance scores computed as $1/(1 + \text{MSE})$. Higher values indicate better performance. Cell color intensity increases with performance, from white (lower score) to darker green (higher score).

5. Discussion

The experimental results highlight several insights regarding both the effectiveness of the proposed aggregation strategies and the broader implications of modeling gender stereotype as a category-structured phenomenon.

Unlike the zero-shot aggregations strategies, which provide a simple and intuitive way to map category-wise binary signals to a final stereotype score, the fine-tuned aggregation detects gender stereotype by learning how to combine category-wise signals.

Across both base and augmented settings, configurations that rely on continuous category scores without thresholding tend to perform better, suggesting that preserving the full range of LLM-generated scores allows the model to capture more nuanced signals of stereotype, rather than losing information through aggressive binarization or thresholding.

The use of augmented data produces mixed results. While augmentation slightly improves performance in some configurations, it does not consistently outperform the best base models. This suggests that, although augmentation increases data diversity, the quality and alignment of augmented examples with real annotator behavior remain critical factors.

Beyond raw performance, the results support the underlying theoretical motivation of this work. By decomposing gender stereotype into interpretable categories, the model is encouraged to reason explicitly about different manifestations of stereotypes rather than collapsing them into a single undifferentiated signal. The fact that supervised aggregation over these category-wise signals yields strong results indicates that such a decomposition is not only conceptually meaningful, but also practically useful.

Importantly, the learned aggregation weights implicitly reflect the relative contribution of different stereotype categories as perceived by human annotators. This aligns with the idea that gender stereotype

is a multidimensional and graded phenomenon, where different dimensions interact and contribute unevenly to the final judgment.

6. Conclusions

This work investigated gender stereotype detection through a structured and category-aware framework that combines LLM-based reasoning with supervised aggregation. Instead of modeling gender stereotype as a single monolithic signal, the proposed approach decomposes it into six interpretable stereotype categories and studies how these dimensions contribute to human judgments of stereotype.

Simple zero-shot aggregation strategies provide a straightforward and interpretable way to combine category-level scores, offering an intuitive baseline. Building on this, we hypothesize that there may be a more nuanced interplay between the individual scores generated by the model, and that learning how these different dimensions interact could lead to more accurate and informative assessments. To explore this idea, we experiment with a lightweight supervised aggregation model that takes as input the raw category scores, allowing the system to learn patterns in how different stereotype dimensions jointly contribute to the overall assessment.

The findings support the theoretical view that gender stereotype is multidimensional, context-dependent, and unevenly distributed across different stereotype categories. By explicitly modeling these dimensions and learning their aggregation from data, the proposed framework offers a flexible and interpretable alternative to end-to-end stereotype classifiers. This design allows the system to adapt to different annotation schemes or definitions of stereotype, while maintaining transparency in how individual stereotype dimensions influence the final outcome.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5 and Gemini-3 in order to conduct grammar and spelling check. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

- [1] L. Arcuri, M. R. Cadinu, *Gli stereotipi Dinamiche psicologiche e contesto delle relazioni social*, Il Mulino, Bologna, 1998.
- [2] S. Cavagnoli, F. Dragotto, et al., *Sessismo*, Mondadori Education, 2021.
- [3] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, A. T. Kalai, Man is to computer programmer as woman is to homemaker? debiasing word embeddings, *Advances in neural information processing systems* 29 (2016).
- [4] A. Caliskan, J. J. Bryson, A. Narayanan, Semantics derived automatically from language corpora contain human-like biases, *Science* 356 (2017) 183–186.
- [5] K. Stanczak, I. Augenstein, A survey on gender bias in natural language processing, *arXiv preprint arXiv:2112.14168* (2021).
- [6] A. T. Cignarella, A. Giachanou, E. Lefever, A survey on stereotype detection in natural language processing, *ACM Computing Surveys* 58 (2025) 1–33.
- [7] Y. Li, G. Zhang, H. Hong, Y. Wang, C. Lin, Overview of the nlpcc 2025 shared task: Gender bias mitigation challenge, in: *CCF International Conference on Natural Language Processing and Chinese Computing*, Springer, 2025, pp. 453–464.
- [8] T. Davidson, D. Warmesley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: *Proceedings of the international AAAI conference on web and social media*, volume 11, 2017, pp. 512–515.

- [9] M. Sap, D. Card, S. Gabriel, Y. Choi, N. A. Smith, The risk of racial bias in hate speech detection, in: *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1668–1678.
- [10] M. Zakizadeh, M. T. Pilehvar, Blind men and the elephant: Diverse perspectives on gender stereotypes in benchmark datasets, *arXiv preprint arXiv:2501.01168* (2025).
- [11] S. Frenda, G. Abercrombie, V. Basile, A. Pedrani, R. Panizzon, A. T. Cignarella, C. Marco, D. Bernardi, Perspectivist approaches to natural language processing: a survey, *Language Resources and Evaluation* 59 (2025) 1719–1746.
- [12] B. Chulvi, L. Fontanella, R. Labadie-Tamayo, P. Rosso, et al., Social or individual disagreement? perspectivism in the annotation of sexist jokes, in: *CEUR Workshop Proceedings*, volume 3494, 2023.
- [13] G. Comandini, M. Speranza, S. Brenna, D. Testa, S. Cavagnoli, B. Magnini, Gsi:detect at evalita 2026: Overview of the task on detecting gender stereotypes in italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [14] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [15] T. Labruna, A. Lavelli, B. Magnini, Testing chatgpt for stability and reasoning: A case study using italian medical specialty tests, in: *Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2023)*, 2023.
- [16] T. Labruna, S. Brenna, G. Bonetta, B. Magnini, Are you a good assistant? assessing llm trustability in task-oriented dialogues, in: *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, 2024, pp. 470–477.
- [17] T. Labruna, S. Gallo, G. Da San Martino, Positional bias in binary question answering: How uncertainty shapes model preferences, in: *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025)*, 2025, pp. 550–560.
- [18] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems* 35 (2022) 24824–24837.
- [19] T. Labruna, G. Bonetta, B. Magnini, Task-oriented dialogue systems through function calling, *RANLP 2025* (2025) 614.
- [20] J. Tavarez-Rodríguez, F. Sánchez-Vega, A. Rosales-Pérez, A. P. López-Monroy, Better together: Llm and neural classification transformers to detect sexism, *Working Notes of CLEF* (2024).
- [21] A. Modzelewski, W. Sosnowski, T. Labruna, A. Wierzbicki, G. Da San Martino, Pcot: Persuasion-augmented chain of thought for detecting fake news and social media disinformation, in: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 24959–24983.

MultiPRIDE at EVALITA 2026: Overview of the Multilingual Automatic Detection of Slur Reclamation in the LGBTQ+ Context Task

Chiara Ferrando^{1,*†}, Lia Draetta^{1,*†}, Marco Madeddu^{1,*†}, Mae Sosto^{2,*†}, Viviana Patti¹, Paolo Rosso^{3,4}, Cristina Bosco¹, Jacinto Mata⁵ and Estrella Gualda⁶

¹University of Torino, Computer Science Department, Corso Svizzera 185, 10149 Torino, Italy

²Human Center Data Analytics Group, Centrum Wiskunde & Informatica, Science Park 123, 1098 XG Amsterdam, The Netherlands

³Pattern Recognition and Human Language Technology Research Center, Universitat Politècnica de València, Camí de Vera, 46022 València, Spain

⁴Valencian Graduate School and Research Network of Artificial Intelligence, Camí de Vera, 46022 València, Spain

⁵Departamento de Tecnologías de la Información, Universidad de Huelva, Campus El Carmen Avda. de las Fuerzas Armadas. - 21007 Huelva, Spain

⁶Departamento de Sociología, Trabajo Social y Salud Pública, Universidad de Huelva, Campus El Carmen Avda. de las Fuerzas Armadas. - 21007 Huelva, Spain

Abstract

MultiPRIDE is a shared task proposed at EVALITA 2026. The main focus of the task is the detection of slur reclamation, a linguistic phenomenon in which derogatory terms are reclaimed by members of a community. The challenge is divided in two main tasks: (A) binary detection of reclamation in social media posts and (B) contextual detection of reclamation considering users' biographies provided within the texts. As we propose a multilingual corpus, each task is further divided into language-specific subtasks (Italian, Spanish and, English). The task has been one of the most participated at EVALITA 2026 [1] with 19 teams submitting a total of 61 runs for Task A and 27 runs for Task B. In this paper, we present the dataset, the evaluation, the participating systems, and a discussion of the results.

All the code is available at <https://github.com/multipride-evalita/MultiPRIDE-Task>

Keywords

Hate Speech, Slur Reclamation, LGBTQ+, Social Media, Multilingual

Warning: This paper contains examples of explicitly offensive content.

1. Motivation

Hate Speech (HS) is intrinsically context-dependent and resists rigid classification, as abusive language can manifest in diverse forms, including insults, hostility, offensiveness, or disrespect [2, 3, 4].

A particularly underexplored dimension of HS concerns the **reclamation of slurs**, a linguistic practice whereby historically derogatory terms are reclaimed to express a sense of belonging and solidarity among in-group members [5, 6].

From a content moderation perspective, accurately identifying reclaimed uses of slurs is crucial to avoid further marginalizing communities that employ such terms as expressions of identity and belonging. Misclassification may restrict their freedom of expression [7, 8, 9] and lead to the paradoxical outcome of censoring forms of counter-speech [10].

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ chiara.ferrando@unito.it (C. Ferrando); lia.draetta@unito.it (L. Draetta); marco.madeddu@unito.it (M. Madeddu); mae.sosto@cw.nl (M. Sosto); viviana.patti@unito.it (V. Patti); proso@dsic.upv.es (P. Rosso); cristina.bosco@unito.it (C. Bosco); mata@uhu.es (J. Mata); estrella@uhu.es (E. Gualda)

ORCID 0009-0000-9593-2510 (C. Ferrando); 0009-0004-6479-5882 (L. Draetta); 0009-0004-5620-0631 (M. Madeddu); 0009-0000-6252-2275 (M. Sosto); 0000-0001-5991-370X (V. Patti); 0000-0002-8922-1242 (P. Rosso); 0000-0002-8857-4484 (C. Bosco); 0000-0001-5329-9622 (J. Mata); 0000-0003-0220-2135 (E. Gualda)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

To the best of our knowledge, slur reclamation remains largely underexplored in Natural Language Processing (NLP) [11], with only Draetta et al. [12] addressing the phenomenon beyond English. Against this background, the **MultiPRIDE task at EVALITA 2026** introduces a multilingual challenge aimed at distinguishing derogatory uses of slurs from their reclaimed uses in LGBTQ+ contexts. The proposed task seeks to advance research on this nuanced aspect of hate speech detection by fostering participation in the identification of reclaimed uses of slurs in a multilingual setting (English, Italian, and Spanish). To this end, we introduce the first multilingual corpus specifically dedicated to slur reclamation within the LGBTQ+ community, comprising social media texts in three languages (English, Italian, and Spanish) and drawn from multiple resources [13, 11, 14]. The dataset includes both social media posts and author biographies for the Italian and Spanish subsets. By incorporating biographical information, the task extends beyond text-level classification to investigate whether user-level information can support the detection of slur reclamation, a phenomenon that is inherently pragmatic and highly subjective. Finally, the shared task enables a systematic analysis of cross-linguistic and cultural variation in slur reclamation practices and facilitates the analysis of how models operating in different languages vary in their ability to address this challenge.

2. Definition of the Task

The task aims to encourage participants to investigate linguistic phenomena and challenges associated with reclaimed language within the LGBTQ+ community. In particular, participants are invited to analyze both the textual properties of the input data (such as argumentative strategies, slurs, denigratory expressions, self-referential labels, and figures of speech), and the contextual information that may be inferred from users’ profiles (when available), including indicators of LGBTQ+ community membership or political orientation. We propose a **binary classification task** in which systems are required to determine whether a term related to the LGBTQ+ domain, as it appears in a sentence, is used with a reclamatory intent or not. The shared task is structured into two main tasks, each further divided into subtasks:

Task A - Slur Reclamation Detection (Textual Message Content Only) In Task A, which consist of binary detection of slur reclamation, participants are provided exclusively with the textual content of the social media posts. Approaches may rely solely on the supplied training data or may incorporate external resources, such as additional annotated datasets. In the latter case, the use of additional data had to be explicitly declared in participants’ technical reports.

Task A is divided into three language-specific subtasks. Participants may choose to address one, two, or all three languages:

- **Subtask A1. *Italian*:** classification on Italian texts;
- **Subtask A2. *Spanish*:** classification on Spanish texts;
- **Subtask A3. *English*:** classification on English texts.

Task B - Contextual Slur Reclamation Detection Task B extends Task A by allowing participants to exploit information associated with the author’s profiles, such as biographical descriptions, when available, in addition to the textual data. Task B includes two language-specific subtasks:

- **Subtask B1. *Italian*:** slur reclamation on Italian texts with contextual metadata;
- **Subtask B2. *Spanish*:** slur reclamation on Spanish texts with contextual metadata;

User biographical information was available only for Spanish and Italian. This information was not present in the English source resource.

Lang	User	Bio	Tweet	Reclamation
it	@user	Certo le circostanze non sono favorevoli.   	In quanto disabile e frocia questi sono i miei PrideMonths. Ma vorrei anche dire che il giorno in cui nel manifesto di un evento lgbtqi+transfemminista verrà citato anche l'antiabilismo oltre ad anti sessismo/obtfobia/razzismo/specismo offro da bere	yes
it	@user	I veri partigiani furono i primi sovranisti! w la patria!	Ecco, adesso pensate all'iter o in affitto ed al male che fate al bambino branco di finocchi arcobaleno	no
es	@user	Me llaman feminista, roja y bollera.   	Buenas tardes a rojos, feministas, republicanos, maricones, bolleras y demás LGTBI  #LGTBI #pride	yes
es	@user	I live for that energy!	Hace rato pasó una caravana de movimiento LGT...etc y algunos me vieron parado observando y me gritaron que yo también era marica. Órale, bien «respetuosas» estas personas que exigen respeto. #PrideMonth #Pride2022 #LGTBQ	no
en	N/A	N/A	I use the word tranny all the time...but that's only in reference to working on my cars. Transgendered, transvestite, and drag queen folk are too fabulous to have their descriptions abbreviated.	yes
en	N/A	N/A	Actually that's what s faggot is. Fag is just something that needs to be burnt.	no

Table 1

Reclaimed and not-reclaimed data examples from the multilingual dataset.

3. Dataset

For the purposes of this challenge, we constructed a multilingual dataset by combining existing publicly available resources in three languages: Italian, Spanish, and English. The data were collected from social media platforms between 2020 to 2022. Specifically, the dataset builds upon the following resources: TWITA, a large-scale Twitter (now X) corpus for Italian spanning over ten years [13]; the LGBTQI+ Dataset (2020-2022)_es, which contains Spanish tweets related to LGBTQ+ topics [14]; and the Reclaimed Hate Speech Dataset (RHSD) for English [11], from which only Twitter and Reddit content were retained.

To obtain a comparable multilingual collection, we applied a uniform multi-step filtering strategy across the three languages. First, we performed keyword-based filtering using homosexuality-related terms drawn from Hurltlex (OM category), a multilingual lexicon of hate speech [15]. This list was subsequently expanded through discussions within a multidisciplinary research team and consultations with the LGBTQ+ community (e.g. *frocio* in Italian, *faggot* in English, and *maricón* in Spanish).

In a second filtering phase, we refined the selection by targeting sentences with a higher likelihood of reclaimed language, identified through positive terms typically expressing pride and community belonging, such as *pride*, *rainbow*, *queer*, and related translations in each language. The complete list of keywords used for all three languages is reported in Appendix A, Table 7.

Table 1 shows some reclaimed and non-reclaimed examples of social media posts.

The dataset, named **MULTIRECLAIM**, comprises a total of 4,983 social media texts: 1,811 in Italian,

1, 711 in English, and 1, 461 in Spanish. For the Italian and Spanish subsets, user biographical information is available, whereas such metadata is not provided in the English source resource. Given the highly context-dependent nature of slur reclamation, biographical information can support annotators by offering additional contextual cues and enabling considerations related to speaker legitimacy.

The **MULTIRECLAIM** dataset was manually annotated by 165 native-speakers, comprising 30 Italian, 70 English, and 65 Spanish speakers. Each text was annotated by five annotators, with efforts made to balance annotator groups by gender and LGBTQ+ self-identification. Annotation was conducted through a dedicated interface, with participants recruited via Prolific crowd-sourcing platform¹ and direct recruitment. Sociodemographic information, including gender, sexual orientation, age, and country of birth, was collected during the process. Annotators were compensated above the U.S. federal minimum wage, estimating an average rate of £9.60 per hour. Participation was voluntary, and individuals could withdraw at any time without consequences.

Human annotators were engaged in a comprehensive annotation process based on a multi-label, fine-grained schema including the identification of offensiveness, irony, and stereotypes. For the purposes of the present task, we exclusively focus on the dimension of slur reclamation. Specifically, annotators were asked to answer the question: *Do you think the terms related to the LGBTQ+ context present in the sentence are used in a reclaimed way?*, thereby performing a binary classification task (yes/no).

The annotations were aggregated with majority voting. We report the label distribution in Table 2, alongside the Inter-Annotator Agreement calculated with Krippendorff’s α .

Language	Positives	Negatives	Total	Krippendorff’s α
Italian	346	1,465	1,811	0.6037
Spanish	221	1,240	1,461	0.3008
English	147	1,564	1,711	0.1553

Table 2

Label distribution and Krippendorff’s α for each language.

During the MultiPRIDE challenge, each language-specific dataset was released separately and split into training and test sets. The training set (60%) was fully annotated and made available to participants under a Creative Commons Attribution 4.0 license, which defines terms for data usage and citation. The test set (40%) was released after the evaluation phase.

4. Evaluation

We provide a separate ranking for each subtask. This implies that teams that created a multilingual approach are still evaluated on each language separately. We also create a baseline for all subtasks.

4.1. Evaluation Metrics

We calculate Precision, Recall and, F1-score for all runs and the two binary labels.

$$Precision_{class} = \frac{\#correct_{class}}{\#assigned_{class}}$$

$$Recall_{class} = \frac{\#correct_{class}}{\#total_{class}}$$

$$F1_{class} = 2 \frac{Precision_{class} * Recall_{class}}{Precision_{class} + Recall_{class}}$$

The final ranking solely based on **F1 macro-score**. As the tasks are all binary, the F1 macro is defined as:

$$F1_{macro} = (F1_{reclaimed} + F1_{not-reclaimed})/2$$

¹<https://www.prolific.com/>

We also report recall, precision and F1-score for each label to better understand differences in performance between the various approaches.

Initially, we aimed to distinguish between *constrained* and *unconstrained* runs. However, persistent ambiguity around what qualifies as constrained made it difficult to apply this distinction consistently. This issue is largely driven by the fact that state-of-art Large Language Models are pre-trained on data that may incorporate information from annotated corpora, lexicons, and other resources. To avoid inconsistent or potentially misleading categorization, we ultimately chose not to label runs as constrained or unconstrained.

4.2. Baselines

For all subtasks, we used the same approach to create the baseline: a fine-tuned language specific BERT model. Specifically, the pre-trained models we chose are: BERT base Italian from MDZ Digital Library for Italian², BETO³ [16] for Spanish, and RoBERTa base for English⁴ [17]. We report all hyperparameters used for the baseline in Appendix B.

The only additional tweak we applied is to add class weights during loss calculation to contrast the severe class-imbalance between positive and negative labels. Also, for Task B, we include the bio by concatenating it to the text and apply different `TYPE_TOKEN_IDS`⁵ to help the models distinguish the two separate information.

We voluntarily chose to have encoder-based baselines to compare them with LLMs which are decoder-based. Our aim is to effectively understand if there is a significant shift from the state-of-the-art performance of the LLMs for text classification or if their improvement is marginal compared to smaller and faster models.

5. Participants and Results

A total of 19 teams, comprising approximately 50 researchers, participated in the MultiPRIDE task. The majority of participants were affiliated with academic institutions, while a small number were independent researchers without formal institutional affiliations. The participating teams exhibited substantial geographical diversity, representing countries across Europe, Asia, and Latin America. European contributions originated from Italy, Spain, France, the Netherlands, and Romania, with Italy being the most represented country. Asian teams were based in Japan, Vietnam, and Iran, while Latin American participation was represented by Mexico.

Participants were allowed to submit systems in three languages—Italian, Spanish, and English—and, for each selected language, to compete in Subtask A, Subtask B, or both. For each language–task combination, teams could submit up to two runs. Table 3 provides an overview of the submissions across languages, tasks, and methods for all participating teams. One of the 19 teams, OUTLIERS, did not submit a final description report.

5.1. Methods

Fine-tuning LM Fine-tuning LM-based models was the dominant strategy, adopted by 17 out of 19 participating teams. Most approaches relied on multilingual or language-specific Transformer architectures. Several teams employed multilingual models such as XLM-RoBERTa or mDeBERTa (e.g., NAMDANG, LLANA, THE HATE BUSTERS), while others leveraged lightweight or parameter-efficient alternatives. Notably, I2C combined an LLM-based system fine-tuned via QLoRA on Qwen2.5-0.5B with a multilingual XLM-RoBERTa classifier. UNI-BO-FICLIT adopted an ELECTRA-based encoder for binary

²<https://huggingface.co/dbmdz/bert-base-italian-uncased>

³<https://huggingface.co/dccuchile/bert-base-spanish-wwm-cased>

⁴<https://huggingface.co/FacebookAI/roberta-base>

⁵This process is automatically done by the HuggingFace Tokenizer when passing a list shaped accordingly: [bio, text]

Team	Country	Task _{Subtasks}	LM Fine-tuning	Data augmentation	Multilingual	Decoder	Feature Extraction	Ensemble	Knowledge Injection
NetGuardAI [18]	Romania 🇷🇴	A _{1,2,3} , B _{1,2}	✓						
I2C [19]	Spain 🇪🇸	A _{1,2,3}	✓	✓	✓	✓			
NamDang [20]	Vietnam 🇻🇳	A _{1,2,3}	✓	✓	✓				
Challenger [21]	Iran 🇮🇷	A _{1,2,3}	✓	✓					
AIWizards [22]	Italy 🇮🇹	B _{1,2}	✓				✓	✓	
MilaNLP [23]	Italy 🇮🇹	A ₁ , B ₁							✓
Ghavidel-Rajabi [24]	Italy 🇮🇹, Iran 🇮🇷	A _{1,2,3}	✓				✓	✓	
LlaNa [25]	Italy 🇮🇹	A ₁ , B ₁	✓	✓					
UniBO-FICLIT [26]	Italy 🇮🇹	A ₁ , B ₁	✓				✓		
HateItOff [27]	France 🇫🇷, The Netherlands 🇳🇱	A _{1,2} , B _{1,2}	✓				✓		
KIT-TIP-NLP [28]	Japan 🇯🇵	A _{1,2,3}	✓	✓	✓				
Outliers*	Mexico 🇲🇽	B ₂	✓						
SaFe Tweets [29]	Italy 🇮🇹	A _{1,3} , B ₁	✓			✓		✓	
The Hate Busters [30]	Italy 🇮🇹	A _{1,2,3} , B _{1,2}	✓		✓		✓		
GRUPPETTOZZO [31]	Italy 🇮🇹	A _{1,2,3} , B _{1,2}	✓	✓	✓				
Kenji-Endo [32]	Italy 🇮🇹	A ₁ , B ₁	✓			✓			
INFOTEC [33]	Mexico 🇲🇽	A ₂	✓						
DataSummit [34]	Italy 🇮🇹	B _{1,2}	✓				✓		
Avahi [35]	Mexico 🇲🇽	A _{1,2,3} , B _{1,2}			✓	✓	✓		✓

Table 3

Overview of teams, tasks and methods. We marked with * the only team who did not submit a report.

classification, and SaFe TWEETS proposed a two-stage pipeline using DeBERTaV3 with LoRA fine-tuning for initial filtering. Language-specific fine-tuning was also common: GRUPPETTOZZO employed umBERTo for Italian, BERT-base for English, and BETO for Spanish, while INFOTEC fine-tuned Spanish models specialized for Mexican and Latin American variants. Other teams incorporated additional signals, such as user biography information (NETGUARDAI, AIWIZARDS), sentiment and emotion features (DATASUMMIT), or synthetic data generated via back-translation (KIT-TIP-NLP). Finally, while submitting their runs, OUTLIERS only mentioned that they fine-tuned Robertuito (Spanish RoBERTa variant).

Data Augmentation Data augmentation was widely adopted, with 7 teams employing it primarily to mitigate class imbalance in the original dataset. I2C applied an augmentation technique based on Contextual Word Embeddings (CWE) exclusively to the training split. NAMDANG explored few-shot prompting with Large Language Models (LLMs) to generate synthetic samples for each target language. Several teams focused on translation-based augmentation strategies. CHALLENGER implemented a comprehensive back-translation augmentation strategy on the training data using the Google Translate API, whereas LLANA adopted a targeted cross-lingual approach by translating reclamation instances from English and Spanish datasets into Italian to strengthen the minority class. Similarly, KIT-TIP-NLP leveraged GPT-4o-mini for systemic multilingual back-translation, expanding the training corpus while preserving semantic content and class distribution.

In contrast, AIWIZARDS introduced user-level soft labels indicating the likelihood of LGBTQ+ community membership, inferred from tweets and user biographies, to train a BERT-like model to predict community membership by considering latent representations associated with LGBTQ+ identity. Finally, GRUPPETTOZZO combined Full Dataset Augmentation and Balanced Augmentation applied to the

minority class strategies using MarianMT [36] and OPUS-MT-based [37, 38] back-translation pipeline to pipeline to generate semantically consistent paraphrases while introducing lexical and syntactic diversity.

Multilingual We want to underline the fact that part of the teams (I2C, NAMDANG, KIT-TIP-NLP, THE HATE BUSTERS, GRUPPETTOZZO, AVAHI) created a single multilingual system to tackle more subtasks instead of creating a language-specific model. Thus, the difficulty of creating such system should be taken into consideration while looking at the results. Most of these approaches leveraged a multilingual BERT and opted to finetune it on all training splits for the different languages. AVAHI created a single multilingual Knowledge Graph and a Retrieval Augmented Generation (RAG) system that took into account the language of the text to classify.

Decoder Four teams (I2C, SAFE TWEETS, KENJI-ENDO, AVAHI) used a decoder model as the main backbone of their system. This was unexpected due to the attention given to generative models in the last years. Out of those five teams, none used a language specific model but rather leveraged LLMs that support multiple languages and have English as their main language in the pre-training. Only AVAHI used a proprietary model, with others using open-weight models.

Feature Extraction One of the most popular approaches to the task is Feature Extraction. Most teams leveraged features related to the sentiment of the text. In fact, their underlying hypothesis was that negative sentiment correlated to offensive use of the terms and inversely correlated to reclamatory use of the slur. Other features that were used include emotion and syntactic information. We want to underline the efforts of GHAVIDEL-RAJABI [24] which defined new indicators that they call Paradox Score, *Sentiment-Hate Gap* and, *Reclamatory Intent Score*.

Ensemble Part of the teams relied on creating an ensemble of models where different streams were then unified to produce a single classification. SAFE TWEETS leveraged an ensemble of lightweight models of different kinds: logistic-regression, SetFit, and a finetuned BERT-like model. Meanwhile, GHAVIDEL-RAJABI combined two different streams: the first based on embeddings of a LM and, the second based on the feature extraction they propose. AI WIZARDS proposed an interesting approach where they leverage two different encoder-based models. One finetuned on the task to detect the likelihood of an author being part of the LGBTQ+ community and the other finetuned on the detection of reclamation.

Knowledge Injection Two teams adopted knowledge injection strategies. Specifically, MILANNLP investigated retrieval-augmented approaches that leverage external knowledge derived from related hate speech corpora to assess whether such information can improve classification performance. To this end, they developed a retrieval-augmented pipeline using the HODI Subtask A corpus [39] as an external knowledge base. In contrast, AVAHI employed a multilingual Knowledge Graph encoding reclaimed terms in Spanish and English for context injection. In particular, potential reclaimed terms identified via the Knowledge Graph were incorporated into the prompt, thereby providing Claude with explicit signals about the linguistic and semantic content of the message.

Pre-Training from Scratch A single team, KENJI-ENDO pre-trained a BabyLM from scratch. This entails that they used a limited number of tokens in the pre-training. They propose two different approaches: (i) *Kenji-Endo Vanilla* a monolithic BabyLM and (ii) *Kendi-Endo MoE* a mixture of experts. As they participated in multiple EVALITA tasks, they finetuned their models on the specific task. For our task, they submitted a single run finetuning *Kendi-Endo Vanilla* for both Task A and B.

5.2. Results

We report the rankings of Task A and Task B in Table 4 and Table 5 respectively. The complete reports for all runs, which include Precision, Recall and, F1-score, can be found in Appendix C.

5.3. Subtask A

For Subtask A, the dominant approach consisted of transformer-based fine-tuning, either in monolingual or multilingual settings. XLM-RoBERTa emerged as the most widely used backbone (adopted by I2C, NAMDANG, GHAVIDEL-RAJABI, LLANA, SAFE TWEETS, THE HATE BUSTERS, and GRUPPETTOZZO), often in combination with multilingual training data. Several teams explored architectural comparisons, notably GHAVIDEL-RAJABI and NAMDANG, which evaluated both XLM-RoBERTa and mDeBERTa-v3, reaching the first position in Subtask A1 (Italian), with an F1-score of 0.8981, and the top-3 in Subtask A2-A3 (Spanish and English), respectively.

A number of teams relied on language-specific encoders, such as ALBERTo (MILANLP), UmBERTo (GRUPPETTOZZO), BETO for Spanish (GRUPPETTOZZO), or ELECTRA (UNIBO-FICLIT), for Italian, and BILMALAT and MEX_Large (INFOTEC) for Spanish. Notably, THE HATE BUSTERS and DATASUMMIT applied Logistic Regression and SVM classifiers, where the first group obtained the best F1-score in Subtask A2 (Spanish). Lastly AVAHI, who outperformed the Subtask A3 (English), adopted a multilingual Knowledge Graph and Claude 4.5 Haiku for final classification.

5.4. Subtask B

Subtask B generally followed similar modeling choices to Subtask A, but included a context integration step, most commonly by concatenating tweet text with user biography or profile information (strategy adopted by NETGUARDAI, MILANLP, GRUPPETTOZZO, AI WIZARDS, and more). Transformer fine-tuning remained the predominant approach, with XLM-RoBERTa and language-specific BERT variants again being the most frequently used architectures (I2C, LLANA, SAFE TWEETS, THE HATE BUSTERS). Within the mentioned groups, LLANA obtained the highest F1-score of 0.9021 for Subtask B1 (Italian).

6. Discussion

6.1. Language Differences

Generally, we can observe that the Italian subtask proved to be easier than other languages with F1-scores hovering around the 0.9 threshold. We hypothesize that this is due to multiple factors that we mainly break down into:

- **Number of Positive Examples and Agreement:** the Italian sub-portion of the training corpus is the one with the highest number of *reclaimed* examples. Also, it is the language with the highest agreement between annotators, 0.6 according to Krippendorff’s α . As both the IAA and model performance is higher for Italian, we can hypothesize that reclamation signals were easier to understand, leading to an easier task.
- **Slurs Distribution:** analyzing the data, we found that Italian had big discrepancies in use for certain slurs. In fact, specific slurs were almost always used in a *reclamatory* manner, meanwhile, others were almost always used in a *non-reclamatory* manner. We report the data regarding each slur for each language in Table 6. In fact, we can see that the word *trans* made up a huge percentage of negative examples. Meanwhile, *forcio*, *forcia* and, *forci* are exclusively used in positive examples, rendering the label easier. We can observe similar patterns of distributions for Spanish and English but not to this extreme.

Team	Run	F1-score	Team	Run	F1-score	Team	Run	F1-score
Ghavidel-Rajabi	1	0.8981	The Hate Busters	2	0.7776	Avahi	1	0.6416
MilaNLP	1	0.8959	NamDang	2	0.7707	SaFe Tweets	1	0.6329
Ghavidel-Rajabi	2	0.8909	NamDang	1	0.7590	NamDang	1	0.6305
SaFe Tweets	1	0.8895	infotec	2	0.7569	Avahi	2	0.6225
LlaNa	1	0.8835	GRUPPETTOZZO	2	0.7506	Challenger	1	0.6204
GRUPPETTOZZO	1	0.8834	HateltOff	1	0.7370	GRUPPETTOZZO	2	0.5979
Challenger	1	0.8816	KIT-TIP-NLP	1	0.7312	NamDang	2	0.5901
HateltOff	1	0.8809	GRUPPETTOZZO	1	0.7289	baseline	1	0.5760
GRUPPETTOZZO	2	0.8735	Avahi	1	0.7117	I2C	2	0.5708
baseline	1	0.8731	I2C	2	0.7105	The Hate Busters	2	0.5500
UniBO-FICLIT	1	0.8707	The Hate Busters	1	0.7098	NetGuardAI	1	0.5476
NamDang	2	0.8589	infotec	1	0.7034	Ghavidel-Rajabi	1	0.5460
HateltOff	2	0.8584	KIT-TIP-NLP	2	0.7016	The Hate Busters	1	0.5291
NetGuardAI	1	0.8537	baseline	1	0.7000	GRUPPETTOZZO	1	0.5285
The Hate Busters	2	0.8503	Ghavidel-Rajabi	1	0.6960	Ghavidel-Rajabi	2	0.5275
I2C	2	0.8435	Challenger	1	0.6774	I2C	1	0.4945
NamDang	1	0.8407	NetGuardAI	1	0.6476	KIT-TIP-NLP	2	0.4545
Kenji-Endo	1	0.8360	HateltOff	2	0.6442	KIT-TIP-NLP	1	0.3973
The Hate Busters	1	0.8345	I2C	1	0.4892			
MilaNLP	2	0.8306						
KIT-TIP-NLP	2	0.8103						
Avahi	1	0.7904						
KIT-TIP-NLP	1	0.7659						
I2C	1	0.6202						

(a) Results for Subtask A1 (Italian)

(b) Results for Subtask A2 (Spanish)

(c) Results for Subtask A3 (English)

Table 4

F1 Macro-Average Scores for Task A

Team	Run	F1-score	Δ with Task A	Team	Run	F1-score	Δ with Task A
LlaNa	1	0.9021	+0.0186	Outliers	1	0.7378	-
baseline	1	0.8981	+0.0250	GRUPPETTOZZO	1	0.7304	+0.0015
GRUPPETTOZZO	1	0.8979	+0.0145	The Hate Busters	1	0.7196	+0.0097
The Hate Busters	2	0.8827	+0.0324	The Hate Busters	2	0.7195	-0.0581
MilaNLP	1	0.8827	-0.0132	Avahi	1	0.7006	-0.0111
GRUPPETTOZZO	2	0.8681	-0.0054	HateltOff	1	0.7005	-0.0365
UniBO-FICLIT	1	0.8644	-0.0063	baseline	1	0.6966	-0.0034
MilaNLP	2	0.8641	+0.0335	AIWizards	2	0.6950	-
AIWizards	1	0.8564	-	GRUPPETTOZZO	2	0.6857	-0.0649
AIWizards	2	0.8549	-	HateltOff	2	0.6733	+0.0291
HateltOff	2	0.8462	-0.0122	DataSummit	1	0.6361	-
The Hate Busters	1	0.8324	-0.0020	AIWizards	1	0.5935	-
HateltOff	1	0.8319	-0.0490	NetGuardAI	1	0.4927	-0.1549
SaFe Tweets	1	0.8164	-0.0731				
Avahi	1	0.8031	+0.0127				
DataSummit	1	0.7734	-				
Kenji-Endo	1	0.7489	-0.0871				
NetGuardAI	1	0.7405	-0.1132				

(a) Results for Subtask B1 (Italian)

(b) Results for Subtask B2 (Spanish)

Table 5

F1 Macro-Average Scores for Task B. For teams that have submitted runs for both Task A and B, we include the difference between the two F1 Macro.

Slur	Pos.	Neg.
checca	2	4
culattone	0	3
finocchi(o)	10	14
finocchietto	0	1
forci(o)	32	0
forci	25	0
forzia	4	0
froci(o)	43	105
frocchia/e	12	3
lella	8	0
ricchione/i	14	15
travestita/e	0	2
travestiti/o	1	22
trans	5	451

(a) Label Distribution for Slurs in Italian

Slur	Pos.	Neg.
bollera(s)	10	12
bolos	0	1
gay(s)	12	89
marica(s)	24	31
maricón(es)	38	97
queer	5	242
trans	9	185
travestis	6	2

(b) Label Distribution for Slurs in Spanish

Slur	Pos.	Neg.
bitch	0	52
drag	2	11
fag	5	22
faggot(s)	48	422
gay	13	143
queen	0	7
queer	17	104
whore	1	6
slut	0	5
sissy	0	2
trans	4	24

(c) Label Distribution for Slurs in English

Table 6

Label Distribution for Slurs in the Test Set; **Pos** indicates the *reclaimed* instances containing the slur; **Neg.** indicates the *not reclaimed* instances containing the slur.

6.2. Baseline Performance and Encoder Models

As explained in Section 4.2, we deliberately chose encoder models as our baseline because they were the most popular systems in the most recent EVALITA campaigns. In fact, we hypothesized that the major shift of the NLP community towards decoder models would also be felt in our task, thus, the baseline could prove how much LLMs affected the state-of-the-art in classification tasks. Surprisingly, most of the runs we received do not actually incorporate LLMs as the main building block of their system.

The importance of encoder-models is also reflected in their performance. Our proposed baselines proved to be quite challenging in almost every subtask, achieving almost always average position and being outperformed by 0.07 in the worst case. Surprisingly, the baseline came in second place for Subtask B1, with a single team overcoming it by a small margin.

This leads us to wonder about the performance of decoder-models. In fact, only one of the five subtasks has been won by a run that relied on LLMs for classification. Interestingly, said task is A3, which is the only English task and the proposed system relied on a proprietary-model (Claude 4.5 Haiku). This can raise a few questions about decoder models in general:

- Are open-weights LLMs actually reliable for these types of classification tasks?
- Have efforts to develop language-specific decoder models been successful? [40]
- Do proprietary models still favor English despite their efforts towards multilinguality?

We realized that these are massive research questions that we cannot answer by relying on our single task, but we hope that our results can contribute to the broader discussion regarding these topics. In our case, we found that: open-weights models still do not perform reliably; we did not receive any run using a language-specific decoder; and that the only win achieved by a LLM was in the English subtask.

6.3. Use of Contextual Information

The only difference between Task A and Task B is related to the possibility of using contextual information about the user, i.e. their social media biographies. Reasonably, it could be expected that the use of additional information about the identity of the author could help the classification systems. In fact, a previous EVALITA task, Sardistance [41] observed major improvements in the contextual subtask compared to textual-only subtask. Crucially, we expected this to happen in our challenge as reclamation is a phenomenon which heavily depends on the identity of the writer.

For clarity, in Table 5, we report the difference in F1-score between Task A and B for each run submitted for both tasks. In more than half cases, we record a decrease in performance by systems using additional information. In cases where systems improve, the increase in performance is not very substantial (a maximum of +0.03). Interestingly, both LLANA and GRUPPETTOZZO which were two of the best performing teams in Subtask B1 and B2, utilized data augmentation in two different manners. The former included translations of the English subset to Italian and, the latter used back translations.

To better understand the difficulties posed by the subtask, we report the causes listed by the teams which recorded the biggest drops in performance. NETGUARDAI hypothesized that the root of the problem is that biographies often included generic text (e.g., “Music lover”, “Madrid”) which acted as noise. This is also stated by SAFE TWEETS who concatenated bio and texts and found that the signal of the reclamation markers was diluted by the bios. An interesting case is KENJI-ENDO, the team proposed a BabyLM, meaning that they performed a pre-training with limited tokens. They noted that the lack of emojis in their data sources negatively affected the model performance in Subtask B1. In fact, they state “the mean byte-fallback ratio is 0.1149 for biographies compared to 0.0134 for tweet text, and 34.3% of biography entries contain at least one such token (versus 9.9% for tweet text)” [32].

7. Conclusion and Future Works

This paper introduces MultiPRIDE, the first shared task on slur reclamation in a multilingual context. The task aims to identify instances in which slurs lose their derogatory intent and are instead used in positive contexts to express identity and belonging. We analyzed the submissions of 19 participating teams and found that slur reclamation constitutes a highly subjective and complex phenomenon, strongly influenced by cultural and linguistic factors. Moreover, system performance varied substantially across languages, suggesting that models struggle to generalize reclamation patterns beyond language-specific cues. This work also presents several limitations, primarily related to the dataset, including the imbalance between reclaimed and non-reclaimed instances. The limited availability of reclaimed examples contributed to reduced performance and higher variance across systems; however, this imbalance reflects the real-world distribution of the slur reclamation phenomenon in online spaces, as it is a highly specific and relatively rare phenomenon. Additionally, the two-step filtering process may introduce a strong lexical bias, as reclaimed instances are more likely to appear in contexts that explicitly reference pride-related concepts. This creates a risk that models rely on surface-level cues rather than deeper signals of reclamation intent. Against this backdrop, future research should focus on extending the dataset to other languages in order to improve coverage and robustness. Furthermore, the integration of culturally grounded features represents a valuable direction, as cultural context can significantly shape the perception of slur reclamation even within the same language. Additionally, we plan to extend the investigation of slur usage to other communities beyond LGBTQ+ community by incorporating additional terms and considering different intersectional dimensions (such as ethnicity and geographic provenance). This will allow us both to enlarge the dataset and to analyze whether models exhibit performance variations when handling slurs reclaimed by different communities. Finally, another line of future work concerns the use of generative AI to produce synthetic reclaimed data, which could help mitigate data scarcity and support more balanced data distribution and experiments.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT (version 5.1), Claude (Sonnet 4.5) in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

Acknowledgments

The work of Paolo Rosso was in the framework of the ANNOTATE-MULTI2 research project (Grant PID2024-156022OB-C32) funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU. The work of Viviana Patti has been partially supported by the “HARMONIA” project - M4-C2, I1.3 Partenariati Estesi - Cascade Call - FAIR - CUP C63C22000770006 - PE PE0000013 under the NextGenerationEU programme.

References

- [1] F. Cutugno, A. Miaschi, A. P. Apro시오, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for Italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] E. W. Pamungkas, V. Basile, V. Patti, Do You Really Want to Hurt Me? Predicting Abusive Swearing in Social Media, in: N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odiijk, S. Piperidis (Eds.), Proceedings of the Twelfth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2020, pp. 6237–6246. URL: <https://aclanthology.org/2020.lrec-1.765>.
- [3] E. W. Pamungkas, V. Basile, V. Patti, Investigating the role of swear words in abusive language detection tasks, *Language Resources and Evaluation* 57 (2023) 155–188.
- [4] E. Holgate, I. Cachola, D. Preoȃuc-Pietro, J. J. Li, Why swear? analyzing and inferring the intentions of vulgar expressions, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, ACL, Brussels, Belgium, 2018, pp. 4405–4414.
- [5] C. Bianchi, Slurs and appropriation: An echoic account, *Journal of Pragmatics* 66 (2014) 35–44.
- [6] B. Cepollaro, D. L. de Sa, The successes of reclamation, *Synthese* 202 (2023) 205.
- [7] B. Cepollaro, P. Labinaz, M. Fasoli, F. Ervas, T. Piazza, M. Croce, T. Numerico, F. Panzeri, G. Tuzet, F. Domaneschi, L. De Vita, S. Di Paola, M. G. Rossi, J. Menichetti, F. Macagno, Social network e comunicazione, *Sistemi Intelligenti* 2019 (2019) 385–632.
- [8] T. Gillespie, *Custodians of the Internet: Platforms, content moderation, and the hidden decisions that shape social media*, Yale University Press, 2018.
- [9] N. Strossen, *Hate: Why we should resist it with free speech, not censorship*, Oxford University Press, 2018.
- [10] B. Cepollaro, M. Lepoutre, R. M. Simpson, Counterspeech, *Philosophy Compass* 18 (2023) e12890. URL: <https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/phc3.12890>.
- [11] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: Proceedings of the 16th ACM Web Science Conference, 2024, pp. 241–249.
- [12] L. Draetta, C. Ferrando, M. Cuccarini, L. James, V. Patti, ReCLAIM Project: Exploring Italian Slurs Reappropriation with Large Language Models, in: Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), 2024, pp. 335–342.
- [13] V. Basile, M. Lai, M. Sanguinetti, Long-term social media data collection at the university of Turin, in: Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR-WS, 2018, pp. 1–6.
- [14] J. Mata, E. Gualda, A dataset of Spanish tweets on people and communities LGBTQI+ during the COVID-19 pandemic 2020-2022 [LGBTQI+ Dataset 2020-2022_es], A dataset of Spanish tweets on people and communities LGBTQI+ during the COVID-19 pandemic 2020-2022 [LGBTQI+ Dataset 2020-2022_es] (2025).
- [15] E. Bassignana, V. Basile, V. Patti, Hurtlex: A Multilingual Lexicon of Words to Hurt, in: E. Cabrio, A. Mazzei, F. Tamburini (Eds.), Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR Workshop Proceedings, Turin, Italy, 2018, pp. 52–57.

- [16] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: Practical ML for Developing Countries Workshop at ICLR 2020, 2020, pp. 1–10.
- [17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: <http://arxiv.org/abs/1907.11692>. arXiv:1907.11692.
- [18] N. C. J. Le Pollotec, E. S. Apostol, C. O. Truică, NetGuardAI at MultiPRIDE: Multilingual Detection of Reclaimed Language in the MultiPRIDE Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [19] L. Vázquez-Ramos, J. Mata-Vázquez, V. Pachón-Álvarez, I2C at MultiPRIDE: Transformers and Generative LLMs for Multilingual Reclamation Classification, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [20] N. Dang, T. Dang, V. T. Kiet, NamDang at MultiPRIDE: Multilingual Classification of Reclaimed Language in LGBTQ+ Discourse using Transformer-based Models, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [21] H. B. A. Tekanlou, M. Bakhtiyarzadeh, J. Razmara, Challenger at MultiPRIDE: Is It Hate Speech or Reclaimed?, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [22] L. Tedeschini, M. Fasulo, AIWizards at MultiPRIDE: A Hierarchical Approach to Slur Reclamation Detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [23] I. Crescenzi, A. Muti, D. Nozza, MilaNLP at MultiPRIDE: Evaluating Lexical, Transformer, and Retrieval-Augmented Models for Reclaimed Language, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [24] H. Rajabi, F. Ghavidel, K. Ghahremani, Ghavidel-Rajabi at MultiPRIDE: Identity, Toxicity, or Complexity? A Language-Specific Feature Selection Approach to Reclamatory Intent Detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [25] A. Mercurio, G. Talluto, I. Siragusa, R. Pirrone, LlaNa at MultiPRIDE: A fine-tuning approach with cross-lingual augmentation, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [26] S. Casazza, UniBO-FICLIT at MultiPRIDE: Fine-Tuning an ELECTRA-Based Model for the Detection of Italian Reclaimed Slurs, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [27] G. Damo, N. B. Ocampo, HateItOff at MultiPRIDE: Linguistic and Sentiment Cues in Reclaimed LGBTQ+ Slur Detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [28] B. G. HB, M. Ptaszynski, R. Melendez, J. Eronen, KIT-TIP-NLP at MultiPRIDE: Continual Learning with Multilingual Foundation Model, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [29] S. Visconti, F. Manzi, SaFeTweets at MultiPRIDE: From Multi-Model Ensembles to Expert LLMs for Reappropriation Detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [30] A. Ciminelli, G. Corvino, C. Gentili, M. Viviani, The Hate Busters at MultiPRIDE: Automatic Identification of Reappropriated Slurs in Multilingual Social Media, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [31] F. Traina, A. Santoro, G. Greco, I. Siragusa, R. Pirrone, GRUPPETTOZZO at MultiPRIDE: Detecting LGBTQ+ Reclamatory Intent via Context-Aware Transformers, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [32] Scozzaro, Calogero Jerik and Rinaldi, Matteo and Mittone, Gianluca and Stranisci, Marco Antonio, Kenji-Endo: a BabyLM @EVALITA, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [33] J. Gleaves, G. Ruiz, INFOTEC-NLP at MultiPRIDE: Spanish Reclaimed Language Classification, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [34] F. Dingeo, M. Viviani, DataSummit at MultiPRIDE: Multilingual Automatic Detection of Reclamation of Slurs in the LGBTQ+ Context Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [35] T. Alcántara, O. Garcia-Vazquez, J. A. Torres-León, M. Cardoso-Moreno, D. Jimenez, L. Moreno-Mendieta, AVAHI at MultiPRIDE: Multilingual Reclaimed Language Detection via Knowledge Graphs and Retrieval-Augmented Generation, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [36] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, F. Seide, U. Germann, A. F. Aji, N. Bogoychev, A. F. T. Martins, A. Birch, Marian: Fast neural machine translation in C++, in: F. Liu, T. Solorio (Eds.), Proceedings of ACL 2018, System Demonstrations, Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 116–121. URL: <https://aclanthology.org/P18-4020/>. doi:10.18653/v1/P18-4020.
- [37] J. Tiedemann, S. Thottingal, OPUS-MT–Building open translation services for the World, in: Annual Conference of the European Association for Machine Translation, European Association for Machine Translation, 2020, pp. 479–480.
- [38] J. Tiedemann, M. Aulamo, D. Bakshandaeva, M. Boggia, S.-A. Grönroos, T. Nieminen, A. Raganato, Y. Scherrer, R. Vázquez, S. Virpioja, Democratizing neural machine translation with OPUS-MT, Language Resources and Evaluation 58 (2024) 713–755.
- [39] D. Nozza, A. T. Cignarella, G. Damo, T. Caselli, V. Patti, HODI at EVALITA 2023: Overview of the first shared task on homotransphobia detection in italian, in: M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi (Eds.), Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), Parma, Italy, September 7th–8th, 2023, volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023, pp. 1–8. URL: <https://ceur-ws.org/Vol-3473/paper26.pdf>.
- [40] B. Magnini, M. Madeddu, M. Resta, R. Zanolì, M. Cimmino, P. Albano, V. Patti, A Leaderboard for Benchmarking LLMs on Italian, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 636–646. URL: <https://aclanthology.org/2025.clicit-1.61/>.
- [41] A. T. Cignarella, M. Lai, C. Bosco, V. Patti, P. Rosso, Sardistance @ EVALITA2020: overview of the task on stance detection in italian tweets, in: Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online event, December 17th, 2020, volume 2765 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 1–10.

A. Dataset Keywords Filtering

Table 7 reports the full set of keywords used in the first and second filtering stages for each language. These keywords were employed to filter the original datasets and to select the texts included in our final datasets.

Language	Stage	Keywords	Final output
Italian [13]	1° filtering	checca, culattone, finocchio, frocio, travestito, trans, ricchione, stesso sesso, legge164, lella, forcio, forci,...	1,890
	2° filtering	pride, lgbt, orgoglio, queer, arcobaleno, diritti, cultura gay, storia gay, lotta, fobia	
English [11]	1° filtering	fag, bitch, gay, puss, queer, drag, whore, queen, slut, sissy, slay	1,711
	2° filtering	pride, lgbt, queer, proud, rainbow, phobia, rights, human, fight	
Spanish [14]	1° filtering	bollera, bolleras, bollos, marica, maricón, maricon, maricones, queer	1,461
	2° filtering	pride, lgbt, orgullo, arcoíris, fobia, derecho, cultura gay, historia gay, lucha	

Table 7

Keyword-based filtering procedure for dataset construction.

B. Baseline Hyperparameters

We report the hyperparameters for the baseline models in Table 8. We did not conduct any hyperparameter search but rather used standard-values for detection tasks for similar domains.

Hyperparameter	Value
Training Epochs	3
Batch Size	16
Learning Rate	$2e^{-5}$
Seq. Length Task A	240
Seq. Length Task B	480
Seed	13

Table 8

Hyperparameter used for the baseline approaches.

C. Complete Results

We report the complete results which include Precision, Recall and, F1 for all labels in Table 9 (for Subtask A1), Table 10 (for Subtask A2), Table 11 (for Subtask A3), Table 12 (for Subtask B1) and, Table 13 (for Subtask B2).

Team	Run	Neg. Label			Pos. Label			Macro Avg.		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Ghavidel-Rajabi	1	0.9517	0.9744	0.9629	0.8800	0.7914	0.8333	0.9158	0.8829	0.8981
MilaNLP	1	0.9561	0.9659	0.9610	0.8496	0.8129	0.8309	0.9029	0.8894	0.8959
Ghavidel-Rajabi	2	0.9441	0.9795	0.9615	0.8974	0.7554	0.8203	0.9208	0.8675	0.8909
SaFe Tweets	1	0.9412	0.9829	0.9616	0.9115	0.7410	0.8175	0.9263	0.8620	0.8895
LlaNa	1	0.9452	0.9710	0.9579	0.8618	0.7626	0.8092	0.9035	0.8668	0.8835
GRUPPETTOZZO	1	0.9409	0.9778	0.9590	0.8879	0.7410	0.8078	0.9144	0.8594	0.8834
Challenger	1	0.9451	0.9693	0.9570	0.8548	0.7626	0.8061	0.9000	0.8659	0.8816
HateItOff	1	0.9436	0.9710	0.9571	0.8607	0.7554	0.8046	0.9021	0.8632	0.8809
GRUPPETTOZZO	2	0.9521	0.9505	0.9513	0.7929	0.7986	0.7957	0.8725	0.8745	0.8735
baseline	1	0.9294	0.9881	0.9578	0.9314	0.6835	0.7884	0.9304	0.8358	0.8731
UniBO-FICLIT	1	0.9319	0.9812	0.9559	0.8981	0.6978	0.7854	0.9150	0.8395	0.8707
NamDang	2	0.9411	0.9539	0.9475	0.7939	0.7482	0.7704	0.8675	0.8511	0.8589
HateItOff	2	0.9194	0.9932	0.9549	0.9565	0.6331	0.7619	0.9380	0.8131	0.8584
NetGuardAI	1	0.9482	0.9369	0.9425	0.7466	0.7842	0.7649	0.8474	0.8605	0.8537
The Hate Busters	2	0.9757	0.8925	0.9323	0.6667	0.9065	0.7683	0.8212	0.8995	0.8503
I2C	2	0.9303	0.9573	0.9437	0.7951	0.6978	0.7433	0.8627	0.8276	0.8435
NamDang	1	0.9372	0.9420	0.9396	0.7500	0.7338	0.7418	0.8436	0.8379	0.8407
Kenji-Endo	1	0.9272	0.9556	0.9412	0.7851	0.6835	0.7308	0.8561	0.8195	0.8360
The Hate Busters	1	0.9383	0.9334	0.9358	0.7254	0.7410	0.7331	0.8318	0.8372	0.8345
MilaNLP	2	0.9295	0.9454	0.9374	0.7519	0.6978	0.7239	0.8407	0.8216	0.8306
KIT-TIP-NLP	2	0.9123	0.9590	0.9351	0.7798	0.6115	0.6855	0.8461	0.7853	0.8103
Avahi	1	0.9252	0.9078	0.9165	0.6400	0.6906	0.6644	0.7826	0.7992	0.7904
KIT-TIP-NLP	1	0.9012	0.9334	0.9170	0.6695	0.5683	0.6148	0.7853	0.7509	0.7659
I2C	1	0.9020	0.6911	0.7826	0.3442	0.6835	0.4578	0.6231	0.6873	0.6202

Table 9

Detailed Performance for Subtask A1 (Italian). Best scores are highlighted in bold.

Team	Run	Neg. Label			Pos. Label			Macro Avg.		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
The Hate Busters	2	0.9287	0.9437	0.9361	0.6500	0.5909	0.6190	0.7894	0.7673	0.7776
NamDang	2	0.9237	0.9497	0.9365	0.6622	0.5568	0.6049	0.7929	0.7533	0.7707
NamDang	1	0.9229	0.9396	0.9312	0.6203	0.5568	0.5868	0.7716	0.7482	0.7590
infotec	2	0.9372	0.9014	0.9190	0.5421	0.6591	0.5949	0.7396	0.7802	0.7569
GRUPPETTOZZO	2	0.9421	0.8833	0.9117	0.5126	0.6932	0.5894	0.7273	0.7882	0.7506
HateItOff	1	0.9086	0.9598	0.9335	0.6667	0.4545	0.5405	0.7876	0.7072	0.7370
KIT-TIP-NLP	1	0.9501	0.8431	0.8934	0.4583	0.7500	0.5690	0.7042	0.7965	0.7312
GRUPPETTOZZO	1	0.9149	0.9296	0.9222	0.5625	0.5114	0.5357	0.7387	0.7205	0.7289
Avahi	1	0.9794	0.7666	0.8600	0.4082	0.9091	0.5634	0.6938	0.8378	0.7117
I2C	2	0.9038	0.9457	0.9243	0.5846	0.4318	0.4967	0.7442	0.6887	0.7105
The Hate Busters	1	0.9287	0.8652	0.8958	0.4508	0.6250	0.5238	0.6898	0.7451	0.7098
infotec	1	0.8948	0.9759	0.9336	0.7209	0.3523	0.4733	0.8079	0.6641	0.7034
KIT-TIP-NLP	2	0.9606	0.7847	0.8638	0.4022	0.8182	0.5393	0.6814	0.8014	0.7016
baseline	1	0.9562	0.7907	0.8656	0.4023	0.7955	0.5344	0.6793	0.7931	0.7000
Ghavidel-Rajabi	1	0.8953	0.9638	0.9283	0.6400	0.3636	0.4638	0.7677	0.6637	0.6960
Challenger	1	0.9707	0.7344	0.8362	0.3684	0.8750	0.5185	0.6696	0.8047	0.6774
NetGuardAI	1	0.9394	0.7485	0.8331	0.3386	0.7273	0.4621	0.6390	0.7379	0.6476
HateItOff	2	0.8812	0.9698	0.9234	0.6053	0.2614	0.3651	0.7432	0.6156	0.6442
I2C	1	0.8504	0.7203	0.7800	0.1524	0.2841	0.1984	0.5014	0.5022	0.4892

Table 10

Detailed Performance for Subtask A2 (Spanish). Best scores are highlighted in bold.

Team	Run	Neg. Label			Pos. Label			Macro Avg.		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Avahi	1	0.9401	0.9281	0.9341	0.3284	0.3729	0.3492	0.6342	0.6505	0.6416
SaFe Tweets	1	0.9385	0.9265	0.9325	0.3134	0.3559	0.3333	0.6260	0.6412	0.6329
NamDang	1	0.9351	0.9441	0.9396	0.3396	0.3051	0.3214	0.6374	0.6246	0.6305
Avahi	2	0.9368	0.9233	0.9300	0.2941	0.3390	0.3150	0.6155	0.6312	0.6225
Challenger	1	0.9356	0.9281	0.9318	0.2969	0.3220	0.3089	0.6162	0.6251	0.6204
GRUPPETTOZZO	2	0.9351	0.8978	0.9161	0.2381	0.3390	0.2797	0.5866	0.6184	0.5979
NamDang	2	0.9270	0.9537	0.9402	0.2927	0.2034	0.2400	0.6099	0.5785	0.5901
baseline	1	0.9402	0.8291	0.8812	0.1955	0.4407	0.2708	0.5679	0.6349	0.5760
I2C	2	0.9255	0.9329	0.9292	0.2222	0.2034	0.2124	0.5739	0.5681	0.5708
The Hate Busters	2	0.9203	0.9776	0.9481	0.3000	0.1017	0.1519	0.6102	0.5397	0.5500
NetGuardAI	1	0.9240	0.8930	0.9082	0.1625	0.2203	0.1871	0.5432	0.5567	0.5476
Ghavidel-Rajabi	1	0.9199	0.9728	0.9457	0.2609	0.1017	0.1463	0.5904	0.5373	0.5460
The Hate Busters	1	0.9188	0.9217	0.9203	0.1404	0.1356	0.1379	0.5296	0.5287	0.5291
GRUPPETTOZZO	1	0.9179	0.9649	0.9408	0.1852	0.0847	0.1163	0.5516	0.5248	0.5285
Ghavidel-Rajabi	2	0.9178	0.9808	0.9483	0.2500	0.0678	0.1067	0.5839	0.5243	0.5275
I2C	1	0.9155	0.8131	0.8613	0.0930	0.2034	0.1277	0.5042	0.5082	0.4945
KIT-TIP-NLP	2	0.9303	0.5974	0.7276	0.1095	0.5254	0.1813	0.5199	0.5614	0.4545
KIT-TIP-NLP	1	0.9325	0.4633	0.6190	0.1016	0.6441	0.1755	0.5170	0.5537	0.3973

Table 11

Detailed Performance for Subtask A3 (English). Best scores are highlighted in bold.

Team	Run	Neg. Label			Pos. Label			Macro Avg.		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
LlaNa	1	0.9625	0.9625	0.9625	0.8417	0.8417	0.8417	0.9021	0.9021	0.9021
baseline	1	0.9517	0.9744	0.9629	0.8800	0.7914	0.8333	0.9158	0.8829	0.8981
GRUPPETTOZZO	1	0.9562	0.9676	0.9618	0.8561	0.8129	0.8339	0.9061	0.8903	0.8979
The Hate Busters	2	0.9394	0.9795	0.9591	0.8947	0.7338	0.8063	0.9171	0.8567	0.8827
MilaNLP	1	0.9394	0.9795	0.9591	0.8947	0.7338	0.8063	0.9171	0.8567	0.8827
GRUPPETTOZZO	2	0.9278	0.9863	0.9562	0.9216	0.6763	0.7801	0.9247	0.8313	0.8681
UniBO-FICLIT	1	0.9343	0.9710	0.9523	0.8534	0.7122	0.7765	0.8939	0.8416	0.8644
MilaNLP	2	0.9371	0.9659	0.9513	0.8347	0.7266	0.7769	0.8859	0.8462	0.8641
AIWizards	1	0.9367	0.9590	0.9477	0.8080	0.7266	0.7652	0.8723	0.8428	0.8564
AIWizards	2	0.9310	0.9676	0.9490	0.8362	0.6978	0.7608	0.8836	0.8327	0.8549
HateItOff	2	0.9187	0.9829	0.9497	0.8980	0.6331	0.7426	0.9083	0.8080	0.8462
The Hate Busters	1	0.9296	0.9471	0.9383	0.7578	0.6978	0.7266	0.8437	0.8225	0.8324
HateItOff	1	0.9216	0.9625	0.9416	0.8053	0.6547	0.7222	0.8634	0.8086	0.8319
SaFe Tweets	1	0.9009	0.9932	0.9448	0.9494	0.5396	0.6881	0.9251	0.7664	0.8164
Avahi	1	0.9422	0.8908	0.9158	0.6257	0.7698	0.6903	0.7840	0.8303	0.8031
DataSummit	1	0.9117	0.9164	0.9140	0.6397	0.6259	0.6327	0.7757	0.7711	0.7734
Kenji-Endo	1	0.9350	0.8345	0.8819	0.5198	0.7554	0.6158	0.7274	0.7949	0.7489
NetGuardAI	1	0.9280	0.8362	0.8797	0.5127	0.7266	0.6012	0.7204	0.7814	0.7405

Table 12

Detailed Performance for Subtask B1 (Italian). Best scores are highlighted in bold.

Team	Run	Neg. Label			Pos. Label			Macro Avg.		
		Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Outliers	1	0.9022	0.9839	0.9413	0.8140	0.3977	0.5344	0.8581	0.6908	0.7378
GRUPPETTOZZO	1	0.9194	0.9175	0.9184	0.5393	0.5455	0.5424	0.7293	0.7315	0.7304
The Hate Busters	1	0.9329	0.8672	0.8989	0.4634	0.6477	0.5403	0.6982	0.7575	0.7196
The Hate Busters	2	0.9021	0.9638	0.9319	0.6667	0.4091	0.5070	0.7844	0.6864	0.7195
Avahi	1	0.9766	0.7565	0.8526	0.3950	0.8977	0.5486	0.6858	0.8271	0.7006
HateItOff	1	0.8992	0.9517	0.9247	0.5932	0.3977	0.4762	0.7462	0.6747	0.7005
baseline	1	0.9498	0.7988	0.8678	0.4012	0.7614	0.5255	0.6755	0.7801	0.6966
AIWizards	2	0.9119	0.8954	0.9036	0.4639	0.5114	0.4865	0.6879	0.7034	0.6950
GRUPPETTOZZO	2	0.8933	0.9598	0.9253	0.6078	0.3523	0.4460	0.7506	0.6560	0.6857
HateItOff	2	0.8879	0.9718	0.9280	0.6585	0.3068	0.4186	0.7732	0.6393	0.6733
DataSummit	1	0.9051	0.8249	0.8632	0.3409	0.5114	0.4091	0.6230	0.6682	0.6361
AIWizards	1	0.9130	0.7183	0.8041	0.2784	0.6136	0.3830	0.5957	0.6660	0.5935
NetGuardAI	1	0.8537	0.9980	0.9202	0.7500	0.0341	0.0652	0.8019	0.5160	0.4927

Table 13

Detailed Performance for Subtask B2 (Spanish). Best scores are highlighted in bold.

AIWizards at MULTIPRIDE: A Hierarchical Approach to Slur Reclamation Detection

Luca Tedeschini^{1,3,*†}, Matteo Fasulo^{2,†}

¹Villanova.ai S.P.A, Località Sa Illetta, SS 195 KM 2.3, 09123 Cagliari, Italy

²Swiss Data Science Center, ETH Zürich, Andresturm, Andreasstrasse 5, 8092 Zürich, Switzerland

³Department of Computer Science and Engineering (DISI), University of Bologna, Mura Anteo Zamboni 7, 40126 Bologna, Italy

Abstract

Detecting reclaimed slurs represents a fundamental challenge for hate speech detection systems, as the same lexical items can function either as abusive expressions or as in-group affirmations depending on social identity and context. In this work, we address Subtask B of the MultiPRIDE shared task at EVALITA 2026 by proposing a hierarchical approach to modeling the slur reclamation process. Our core assumption is that members of the LGBTQ+ community are more likely, on average, to employ certain slurs in a reclamatory manner. Based on this hypothesis, we decompose the task into two stages. First, using a weakly supervised LLM-based annotation, we assign fuzzy labels to users indicating the likelihood of belonging to the LGBTQ+ community, inferred from the tweet and the user bio. These soft labels are then used to train a BERT-like model to predict community membership, encouraging the model to learn latent representations associated with LGBTQ+ identity. In the second stage, we integrate this latent space with a newly initialized model for the downstream slur reclamation detection task. The intuition is that the first model encodes user-oriented sociolinguistic signals, which are then fused with representations learned by a model pretrained for hate speech detection. Experimental results on Italian and Spanish show that our approach achieves performance statistically comparable to a strong BERT-based baseline, while providing a modular and extensible framework for incorporating sociolinguistic context into hate speech modeling. We argue that more fine-grained hierarchical modeling of user identity and discourse context may further improve the detection of reclaimed language. We release our code at <https://github.com/LucaTedeschini/multipride>.

Keywords

Hate Speech Detection, Reclaimed Slurs, User Profiling, Hierarchical Modeling, Social Media NLP

1. Introduction

The automatic detection of hate speech and abusive language remains a central challenge in natural language processing, particularly in social media contexts where meaning is highly context-dependent. A recurring difficulty arises from the use of slur terms, whose interpretation cannot be determined solely from surface lexical content. In many communities, historically derogatory expressions are re-appropriated and used in neutral, positive, or identity-affirming ways, a phenomenon known as *semantic reclamation* [1, 2]. As a result, identical lexical items may convey either abuse or solidarity depending on pragmatic, social, and author-level factors.

Warning: *This paper contains examples of explicitly offensive content.*

This ambiguity poses a fundamental limitation for text-only classification models. While modern pretrained language models are effective at capturing local linguistic regularities, they often struggle to infer speaker intent when the interpretation of a slur depends on extra-linguistic context. Prior sociolinguistic research highlights that reclaimed usage is strongly associated with in-group membership, self-identification, and shared social knowledge [3, 4].

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ ltedeschini@villanova.ai (L. Tedeschini); mfasulo@ethz.ch (M. Fasulo)

🌐 <https://github.com/LucaTedeschini> (L. Tedeschini); <https://www.matteofasulo.com> (M. Fasulo)

🆔 0009-0006-0375-829X (L. Tedeschini); 0000-0002-7019-3157 (M. Fasulo)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Despite this, most computational approaches to hate speech detection operate under the assumption that such contextual information is either unavailable or implicitly encoded in the text itself. The MultiPRIDE shared task at EVALITA 2026 [5, 6] explicitly addresses this limitation by providing user biographies alongside tweets for Subtask B, enabling models to incorporate author-level context when determining whether a slur is used in a reclamatory or offensive manner.

In this work, we investigate whether decomposing the classification process into two hierarchical levels, assuming a directional dependency between user identity and language use, can improve performance over standard fine-tuning of a BERT-like model.

Our approach is motivated by the hypothesis that there exists a strong association between user identity and the reclamatory use of slurs. In particular, users who self-identify as part of, or closely aligned with, the LGBTQ+ community may be more likely to employ certain slurs in a reclamatory manner compared to users who are not. We further hypothesize that inferring such identity-related signals is comparatively easier when leveraging user biographies, where individuals often explicitly express aspects of their identity through self-descriptions, pronouns, symbols, or community-related markers.

Based on these assumptions, we adopt a hierarchical modeling strategy. First, we train a model to encode users into a latent space capturing signals related to LGBTQ+ self-identification. We then integrate this latent representation with a second model initialized with weights pretrained for hate speech detection. The underlying intuition is that the two models encode complementary perspectives on the input, one emphasizing sociolinguistic user context and the other focusing on abusive language patterns, whose fusion yields a richer representation for downstream slur reclamation detection.

Experiments on Italian and Spanish show that the proposed architecture achieves performance statistically equivalent to a strong BERT-based baseline. While this does not result in an improvement in aggregate evaluation metrics, it establishes a modular framework for incorporating user-level sociolinguistic context into hate speech modeling. We argue that extending this hierarchical approach to incorporate additional contextual signals, such as multimodal content or social network information, may further enhance the detection of reclaimed language in future work.

2. Task and Data

2.1. Task Definition

The shared task addresses slur reclamation detection in social media. Given a tweet containing at least one LGBTQ+ related term, the system must predict whether the term is used in a reclamatory manner (i.e., in-group, non-derogatory, or empowering usage) or in a non-reclamatory manner (e.g., derogatory or otherwise non-reclaiming uses).

The task is divided into two subtasks: Task A, which relies solely on tweet text, and Task B, which additionally incorporates contextual information such as user biographies. Building on the motivation outlined in Section 1, we restrict the scope of our contribution to Task B.

2.2. Data Characteristics

The dataset provided for the challenge exhibits a substantial class imbalance, as summarized in Table 1.

Table 1

Label distribution for the slur reclamation task. *Non-reclamatory* instances correspond to uses of LGBTQ+ related terminology that are derogatory or not employed in a self-affirming or in-group manner.

Language	Non-reclamatory	Reclamatory
Italian	879 (80.95%)	207 (19.05%)
Spanish	743 (84.83%)	133 (15.17%)

Table 2

Representative examples of tweets with their labels and corresponding user biographies. Tweets and biographies are truncated for readability.

Language	Label	Tweet	Biography
Italian	Non-reclamatory	ORA, I CLANDESTINI, AVRANNO SUBITO DIRITTI DI CITTADINANZA ITALIANA [...]	Voglio un governo che tuteli noi italiani
Italian	Reclamatory	lo amo essere frocio, viva i finocchi e tutt* i/le/lə miə amicə ricchionə	Cogito ergo cum – he/him
Spanish	Non-reclamatory	#orgullogay no me jodan puedo sentir orgullo de ser un buen padre [...]	SACERDOTE MAYOR DE LA RELIGIÓN YORUBA [...]
Spanish	Reclamatory	Hoy amanecí más marica que todos los días! [...]	ARQUITECTO – Popayán, Colombia.

For both languages, over 80% of the tweets contain terms used in a non-reclamatory manner. This imbalance necessitates the adoption of mitigation strategies, as the objective is to minimize both false positives (incorrectly flagging reclaimed in-group usage) and false negatives (failing to detect harmful or derogatory uses). Table 2 presents representative examples from the dataset in both Italian and Spanish.

To create training, validation, and test sets, we performed two consecutive stratified splits on the original dataset resulting in an overall 70% / 15% / 15% split for train, validation, and test, respectively. Stratification ensures that each subset maintains the same label distribution as the original dataset, mitigating issues due to class imbalance reported in Table 1. For all experiments, the train/validation/test splits were fixed using a set seed to ensure reproducibility.

2.3. Why User Content Matters

A qualitative inspection of the data suggests that lexical cues alone are often insufficient to resolve reclamation. Tweets are frequently short, elliptical, ironic, or contextually ambiguous, and may include quoted speech or re-contextualized language. In such settings, user self-descriptions provide additional context that can be indicative of stance and community alignment.

Profile text may contain self-disclosed signals, such as pronouns, identity-related descriptors, or community markers, that correlate with in-group usage patterns. In this study, however, we conceptualize these signals strictly as contextual cues derived from text, rather than as veridical or ground-truth indicators of personal identity.

2.4. LLM-Assisted Proxy Labeling for the User Encoder

Building on this intuition, we define an auxiliary proxy variable, referred to as an *LGBTQ+ affiliation signal*, which is used exclusively to train the user encoder. Concretely, we automatically assign a binary label based on the concatenation of a user’s tweet and biography using an instruction-tuned Large Language Model (LLM), specifically DeepSeek-V3.2 [7]. The two segments are concatenated using the separator token [SEP] between the tweet and biography. No additional separator or custom template is employed. The resulting label is intended to capture whether the available textual evidence suggests in-group affiliation or solidarity, rather than to infer or validate a protected attribute.

The exact prompts used for LLM-based annotation are provided in Appendix A. Italian tweets were annotated using the original Italian prompt, while Spanish tweets were annotated using a Spanish translation of the prompt. For transparency and reproducibility, the Appendix includes both versions, with the Spanish prompt directly translated from the Italian text.

We explicitly acknowledge that automatic user profiling raises ethical and methodological concerns, including issues of privacy, consent, and the risk of stereotyping or misclassification. To mitigate these risks, we employ these labels solely as a form of weak supervision for representation learning,

not as a user-level prediction target. The objective is to encourage the model to encode latent social and contextual signals useful for reclamation detection, without claiming accurate identification of individuals or verification of protected attributes. Accordingly, we do not report user-level outcomes and focus exclusively on aggregate task performance.

The resulting auxiliary label distribution is reported in Table 3.

Table 3

Distribution of proxy “LGBTQ+ affiliation” labels across languages. Labels are derived via LLM-assisted weak supervision and used to train the user encoder.

Language	Affiliated	Not Affiliated
Italian	430 (39.60%)	656 (60.40%)
Spanish	362 (41.32%)	514 (58.68%)

Notably, this auxiliary distribution is more balanced than the primary task labels. At the same time, the auxiliary task is only imperfectly aligned with reclamation: while community alignment may increase the likelihood of reclaimed usage, it does not entail it (e.g., in cases of neutral self-reference, descriptive mentions, or internal criticism). This imperfect alignment is beneficial in practice, as it encourages the model to encode broader contextual information rather than collapsing to a narrow lexical heuristic.

3. System Description

We formulate slur reclamation detection as a binary classification task whose interpretation depends on both (i) the local linguistic context of the tweet and (ii) author-level contextual cues derived from user self-descriptions. To model these signals, we adopt a dual-encoder architecture with gated representation-level fusion.

3.1. Baseline Model

As a reference system, we fine-tune a BERT-like encoder using a class-weighted loss function, following the recommendations of the MultiPRIDE shared task organizers. Italian and Spanish are treated independently, using language-specific pretrained models: *nickprock/setfit-italian-hate-speech* [8] for Italian and *cardiffnlp/twitter-xlm-roberta-base* [9] for Spanish.

Each baseline model is trained using identical hyperparameters. To reduce variance due to random initialization, we report results averaged over five runs with different random seeds. The input to the model consists of the concatenation of the tweet text and the corresponding user biography. Performance results are reported in Table 4.

Table 4

Performance of the baseline BERT-based models on the development set, averaged over five random seeds (mean $\pm$ standard deviation).

Language	Accuracy	F1 score
Spanish	0.835 $\pm$ 0.034	0.671 $\pm$ 0.033
Italian	0.942 $\pm$ 0.017	0.904 $\pm$ 0.027

A noticeable performance gap emerges between the two languages, with the Italian baseline already achieving relatively strong scores. This discrepancy may stem from differences in annotation practices, data distributions, model pretraining domains, or language-specific sociolinguistic factors. Overall, the strength of the baseline makes further improvements non-trivial and motivates architectural modifications rather than incremental hyperparameter tuning.

3.2. Dual-Encoder Architecture

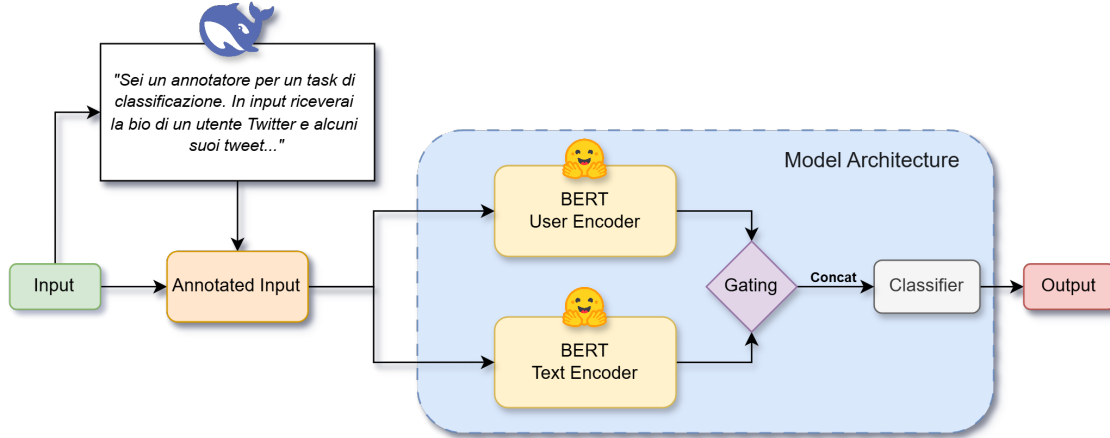


Figure 1: Overview of the proposed dual-encoder architecture. A text encoder and a user encoder process the concatenation of tweet and user biography in parallel. Their final hidden representations are fused through a learned gating mechanism to produce a user-aware representation for slur reclamation classification.

The proposed architecture consists of two parallel BERT-like encoders that process identical inputs, the concatenation of the tweet and the user biography, but are optimized under different training signals. Separate dual-encoder models are constructed for Italian and Spanish, reusing the same language-specific backbones adopted in the baseline.

The key intuition is that the two encoders learn complementary representations: one encoder is optimized directly for slur reclamation detection, while the other is encouraged to encode broader user-related sociolinguistic signals via an auxiliary proxy task. These representations are subsequently combined through a learned gating mechanism.

Although the user encoder also processes the tweet in addition to the biography, this architectural choice was informed by preliminary experiments indicating degraded performance when relying exclusively on the biography. Incorporating both segments enables the user encoder to leverage the full range of contextual signals (including lexical style, syntactic structure, and phrasing patterns) that underlie the LLM-derived proxy labels. Consequently, the text encoder remains dedicated to the primary task, thereby yielding complementary and mutually informative representations.

3.2.1. User Encoder

To induce user-oriented representations, we fine-tune the user encoder using the auxiliary proxy labels previously described. These labels, derived via LLM-assisted weak supervision, reflect whether the combined tweet and biography text suggests in-group affiliation or solidarity, without asserting ground-truth identity.

Performance on the proxy task is reported in Table 5.

Table 5

Performance of the user encoder on the proxy “LGBTQ+ affiliation” task, averaged over five random seeds (mean $\pm$ standard deviation).

Language	Accuracy	F1 score
Spanish	0.792 ± 0.038	0.784 ± 0.041
Italian	0.753 ± 0.021	0.736 ± 0.024

Although this auxiliary task is imperfectly aligned with slur reclamation, the encoder learns latent

representations that capture broader contextual and sociolinguistic cues. When integrated into the dual-encoder architecture, these representations provide complementary information to the text-focused encoder.

We initially experimented with freezing the user encoder during downstream training, under the assumption that preserving its learned representations would prevent task-specific overfitting. However, empirical results consistently showed that allowing the user encoder to be fine-tuned jointly with the rest of the model yielded better performance. Consequently, all final experiments employ fully trainable encoders.

3.3. Training Recipe

3.3.1. Hyperparameters

Both encoders are trained using standard fine-tuning settings. We use a learning rate of 2×10^{-5} , batch size of 8, and weight decay of 0.1. For the dual-encoder model, we adopt a Linear Probing–Fine-Tuning (LPFT) strategy: the fusion and classification layers are first trained with frozen encoders, followed by joint fine-tuning of the entire architecture using a reduced learning rate of 5×10^{-6} .

3.3.2. Loss Function

To address class imbalance, we experimented with Focal Loss [10] as an alternative to standard Cross Entropy. For Spanish, Cross Entropy achieved higher F1 scores than Focal Loss, while for Italian the two losses yielded comparable performance. Given the lack of consistent improvements, Cross Entropy was adopted for all final submissions.

3.3.3. Feature Fusion via Learned Gating

The representations produced by the two encoders are combined using a learned gating mechanism. Let $\mathbf{h}_{\text{text}}$ and $\mathbf{h}_{\text{user}}$ denote the final hidden states of the text and user encoders, respectively. The gating vector $\mathbf{g}$ is computed as:

$$\mathbf{g} = \sigma(\mathbf{W}_2 \cdot \tanh(\mathbf{W}_1 \cdot [\mathbf{h}_{\text{text}} :: \mathbf{h}_{\text{user}}])), \quad (1)$$

where $[\cdot :: \cdot]$ denotes concatenation. The fused representation is then obtained as:

$$\mathbf{h}_{\text{fused}} = \mathbf{g} \odot \mathbf{h}_{\text{text}} + (1 - \mathbf{g}) \odot \mathbf{h}_{\text{user}}, \quad (2)$$

with $\odot$ indicating element-wise multiplication. This mechanism allows the model to dynamically balance reliance on textual content and user-level contextual signals on a per-instance basis.

3.3.4. Training Procedure

The overall training pipeline consists of three stages: (i) fine-tuning a baseline text encoder on the slur reclamation task, (ii) training the user encoder on the auxiliary proxy task to induce user-oriented representations, and (iii) training the fused dual-encoder model on the primary reclamation task, jointly optimizing the gating mechanism, classifier, and both encoders.

4. Results

To provide a comprehensive overview of model performance, we report results on the development set averaged over five runs for both the baseline and the proposed dual-encoder architecture in Table 6. In addition, we report performance on the official test set released by the shared task organizers in Table 7.

Table 6

Development set performance (macro F1) of baseline and dual-encoder models, averaged over five random seeds. p -values indicate statistical significance with respect to the baseline.

Model	Language	F1 (mean $\pm$ std)	p -value
Baseline	Italian	0.90 $\pm$ 0.03	-
Baseline	Spanish	0.67 $\pm$ 0.04	-
Dual Encoder	Italian	0.88 $\pm$ 0.04	0.28
Dual Encoder	Spanish	0.64 $\pm$ 0.02	0.17

Table 7

Official test set performance (macro-averaged) of the dual-encoder model. p -values were not computed for this set.

Language	Precision	Recall	F1-score
Italian	0.88	0.83	0.85
Spanish	0.69	0.70	0.70

4.1. Results Discussion

The results indicate that the proposed dual-encoder architecture achieves performance statistically equivalent to the baseline on the development set ($p > 0.05$), which itself constitutes a strong benchmark for the task. On the official test set, the dual-encoder model maintains performance comparable to the baseline, confirming that introducing additional modeling complexity does not degrade results. Overall, this suggests that the dual-encoder framework successfully incorporates user-level context without harming predictive accuracy.

While the dual-encoder model does not yield improvements in aggregate evaluation metrics, it demonstrates that incorporating user-oriented representations via hierarchical and weakly supervised modeling can be achieved without sacrificing classification accuracy. We view this outcome as a validation of the architectural framework rather than as an endpoint. In particular, the modular nature of the approach allows for future extensions, such as alternative fusion mechanisms, more refined proxy tasks, or the integration of additional contextual encoders, that may more fully exploit user-level information.

4.2. Differences Between Italian and Spanish

The substantial performance gap observed between the Italian and Spanish models is likely attributable to differences in the underlying baseline architectures and pretraining regimes rather than to intrinsic linguistic properties of the two languages.

The Italian baseline relies on a SetFit-based model, which is specifically designed to perform well in low-resource and few-shot settings. Moreover, this model has been previously fine-tuned on a hate speech detection task, which is closely related to slur reclamation detection. Such task-adjacent pretraining plausibly facilitates positive transfer, as both tasks require sensitivity to nuanced, context-dependent uses of potentially offensive language.

In contrast, the Spanish baseline is based on a general-purpose multilingual transformer that does not benefit from comparable task-specific pretraining and is less explicitly optimized for few-shot learning scenarios. As a result, the Spanish model may struggle to generalize effectively under limited supervision. We therefore attribute the observed performance disparity primarily to architectural and pretraining differences, rather than to language-specific sociolinguistic factors.

4.3. Error Analysis

We conclude by examining representative misclassified examples from the Italian dataset. Despite the strong overall performance, the model occasionally fails on instances that contain overtly derogatory slur usage and are unambiguously labeled as non-reclamatory.

For example, the following tweets were incorrectly classified as reclamatory despite exhibiting clear hostile intent:

Caro #Zan, quando froci, ricchioni, pederasti e lesbiche impareranno a rispettare i diritti altrui io smetterò di chiamarli così...
Maledetti Grullini... un'accozzaglia arcobaleno di froci e lesbiche. Possa il Coronavirus fare pulizia tra di voi...
Ronaldo è solo un travestito da gay pride che se la fa con quell'altro effeminato di Dybala

These errors suggest that the model may, in some cases, overestimate the likelihood of reclamatory usage when specific slur terms are present, independently of their surrounding discourse. One possible explanation is that certain lexical items are underrepresented in explicitly non-reclamatory contexts within the training data, leading to biased latent representations. Another contributing factor may be the influence of user-level contextual signals encoded by the user encoder, which could inadvertently dominate the gating mechanism even when the tweet content is clearly hostile.

While the precise cause of these errors cannot be conclusively identified within the scope of this work, they highlight an important limitation of representation-based fusion approaches: contextual signals, if not sufficiently constrained, may override strong local indicators of abuse. Future work could address this issue by introducing explicit regularization mechanisms, contrastive objectives, or token-level supervision to better disentangle reclaimed usage from overtly denigratory language.

5. Conclusion

In this work, we presented a dual-encoder architecture for slur reclamation detection that explicitly integrates textual content with user-level contextual information derived from profile descriptions. We evaluated the approach on the MultiPRIDE dataset for Italian and Spanish, showing that the proposed model achieves strong performance across both languages.

While the dual-encoder architecture does not yield statistically significant improvements over a standard BERT-based baseline (which itself constitutes a highly competitive benchmark) it demonstrates that user-oriented contextual representations can be incorporated without degrading performance. These findings suggest that hierarchical and representation-based modeling of sociolinguistic context is a viable direction for slur reclamation detection.

The primary contribution of this work lies in the architectural framework rather than in immediate performance gains. By explicitly separating text-based and user-oriented representations and combining them through a learned fusion mechanism, we provide a flexible and extensible structure for modeling context-dependent language phenomena.

5.1. Future Work

The strength of the baseline indicates that slur reclamation detection is already well addressed by current pretrained language models when sufficient textual context is available. Nevertheless, the modular design of the proposed dual-encoder architecture opens several avenues for future research.

Future work may focus on developing more expressive fusion strategies for heterogeneous latent spaces, as well as on refining the auxiliary supervision used to induce user-oriented representations. Additionally, the architecture naturally supports the integration of further contextual encoders, such as those operating on social network structures or multimodal inputs (e.g., images). We expect that more effective integration of these diverse information sources may enable models to capture finer-grained distinctions in reclaimed language use and ultimately surpass current strong baselines.

Declaration on Generative AI

During the preparation of this work, the author(s) used Google Gemini 3.0 in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgments

This work was supported by the project “IPCEI-CIS - Progetto Villanova” (Beneficiary: Tiscali Italia S.p.A., Prog. n. SA. 102519 – CUP B29J24000850005).

References

- [1] R. Jeshion, Pride and prejudiced: On the reclamation of slurs, *Grazer Philosophische Studien* 97 (2020) 106 – 137. URL: https://brill.com/view/journals/gps/97/1/article-p106_106.xml. doi:10.1163/18756735-09701007.
- [2] L. Filardo-Llamas, A. Roldán-García, Between “queer” and “cishet men”. lexical creativity, gender identities and sexualities in linguistic labels in tumblr, *Journal of Language and Discrimination* 9 (2025) 1–27. URL: <https://doi.org/10.3138/jld-2025-0501>. doi:10.3138/jld-2025-0501. arXiv:<https://doi.org/10.3138/jld-2025-0501>.
- [3] C. Bianchi, Slurs and appropriation: An echoic account, *Journal of Pragmatics* 66 (2014) 35–44. URL: <https://www.sciencedirect.com/science/article/pii/S0378216614000526>. doi:<https://doi.org/10.1016/j.pragma.2014.02.009>.
- [4] B. Cepollaro, D. López de Sa, The successes of reclamation, *Synthese* 202 (2023). doi:10.1007/s11229-023-04393-y.
- [5] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [6] F. Cutugno, A. Miaschi, A. P. Apro시오, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [7] DeepSeek-AI, Deepseek-v3.2: Pushing the frontier of open large language models, 2025.
- [8] L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, O. Pereg, Efficient few-shot learning without prompts, 2022. URL: <https://arxiv.org/abs/2209.11055>. arXiv:2209.11055.
- [9] F. Barbieri, L. Espinosa Anke, J. Camacho-Collados, XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond, in: *Proceedings of the Thirteenth Language Resources and Evaluation Conference, European Language Resources Association, Marseille, France, 2022*, pp. 258–266. URL: <https://aclanthology.org/2022.lrec-1.27>.
- [10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, 2018. URL: <https://arxiv.org/abs/1708.02002>. arXiv:1708.02002.

A. LLM Annotation Prompts

Italian prompt

Sei un annotatore per un task di classificazione.
In input riceverai la bio di un utente Twitter e alcuni suoi tweet
Il tuo compito è decidere se l'utente in questione fa parte o meno della comunità LGBT.
Basa la tua scelta sul modo di scrivere, sul contenuto della bio e dei tweet, e su tutti i fattori che ritieni rilevanti.
L'input che ti fornirò sarà nel formato TWEET - BIO.

L'output che voglio è semplicemente un numero:
0 se l'utente NON appartiene alla comunità LGBT,
1 se invece appartiene alla comunità.

Esempio di interazione:
INPUT: "fuck gender rules and the rules of society || bts || exo" - "pansexual, genderqueer and polyamorous [LGBTQ+ flag] || she/her || unito dams"
OUTPUT: 1

Spanish prompt

Eres un anotador para una tarea de clasificación.
En la entrada recibirás la biografía de un usuario de Twitter y algunos de sus tuits.
Tu tarea es decidir si el usuario en cuestión forma parte o no de la comunidad LGBT.
Basarás tu decisión en la forma de escribir, el contenido de la biografía y de los tuits, y en todos los factores que consideres relevantes.
La entrada que te proporcionaré tendrá el formato TWEET - BIO.

La salida que quiero es simplemente un número:
0 si el usuario NO pertenece a la comunidad LGBT,
1 si sí pertenece a la comunidad.

Ejemplo de interacción:
ENTRADA: "fuck gender rules and the rules of society || bts || exo" - "pansexual, genderqueer and polyamorous [LGBTQ+ flag] || she/her || unito dams"
SALIDA: 1

Note. The token [LGBTQ+ flag] is a placeholder for the LGBTQ+ flag emoji, encoded by the Unicode sequence U+1F3F3 U+FE0F U+200D U+1F308.

AVAHI at MultiPRIDE: Multilingual Reclaimed Language Detection via Knowledge Graphs and Retrieval-Augmented Generation

Tania Alcántara^{1,2,*,†}, Omar Garcia-Vazquez^{2,†}, Jose A. Torres-León^{1,†},
Marco Cardoso-Moreno^{1,2,†}, Diana Jimenez^{1,2,†} and Luis Moreno-Mendieta^{1,2,†}

¹AVAHI, Mexico, City, 06600, Mexico

²Instituto Politécnico Nacional, Centro de Investigación en Computación, Mexico, City, 07700, Mexico

Abstract

This paper presents a hybrid approach to reclaimed language classification in LGBTQ+ tweets for the MultiPride 2026 shared task. Reclaimed language—historically derogatory terms reappropriated by marginalized communities—requires sophisticated contextual understanding due to its usage-dependent meaning. The system integrates three components: a multilingual Knowledge Graph encoding reclaimed terms across Spanish, English, and Italian; a Retrieval-Augmented Generation system using sentence embeddings for case-based reasoning; and Claude 4.5 Haiku accessed via AWS Bedrock for final classification. The system processes both textual content alone (Task A) and text with author biographical information (Task B). Results show substantial cross-linguistic variation: 33.5% positive predictions for Spanish, 20.7% for Italian, and 9.8% for English. Biographical context provided differential benefits, with Italian showing 14.0% relative improvement and Spanish showing minimal change. Average confidence scores remained high (0.857–0.892) across all configurations. The findings demonstrate the effectiveness of combining symbolic knowledge, neural retrieval, and large language models for context-dependent sociolinguistic classification, while highlighting language-specific differences in the value of biographical context for disambiguation. Warning: This paper contains examples of explicitly offensive content.

Keywords

Reclaimed language, LGBTQ+ linguistics, Retrieval-Augmented Generation, Knowledge graphs, Multilingual classification

1. Introduction

Hate speech targeting the LGBTQ+ community in digital environments has become an increasingly relevant phenomenon due to its profound social, cultural, and psychological implications. Online platforms not only amplify the spread of discriminatory messages but also create spaces where language can be used both as a tool of exclusion and as a form of resistance. In this context, expressions historically considered offensive toward sexual and gender diversity do not always function unambiguously: they are often reclaimed by the LGBTQ+ community itself as forms of identity affirmation, irony, or empowerment.

This phenomenon of linguistic reclamation, in which originally pejorative terms are recontextualized and transformed into expressions of pride, belonging, and resistance, presents a substantial challenge

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ tania.alcantara@avahi.ai, talcantaram2020@cic.ipn.mx (T. Alcántara); ogarciav2023@cic.ipn.mx (O. Garcia-Vazquez); jose.torres@avahi.ai (J. A. Torres-León); marco.cardoso@avahi.ai (M. Cardoso-Moreno); diana.lopez@avahi.ai (D. Jimenez); enrique.moreno@avahi.ai (L. Moreno-Mendieta)

🌐 <https://www.linkedin.com/in/talcantaram/> (T. Alcántara); <https://www.linkedin.com/in/omar-garcia-vazquez-093128219/> (O. Garcia-Vazquez); <https://www.linkedin.com/in/josAI-alberto-torres-00a446140/> (J. A. Torres-León); <https://www.linkedin.com/in/cardoso1994/> (M. Cardoso-Moreno); <https://www.linkedin.com/in/diana-jimenez-aab038201/> (D. Jimenez); <https://www.linkedin.com/in/enrique-mm/> (L. Moreno-Mendieta)

🆔 0009-0006-4619-9352 (T. Alcántara); 0009-0002-1205-1166 (O. Garcia-Vazquez); 0000-0003-2704-0216 (J. A. Torres-León); 0009-0001-1072-2985 (M. Cardoso-Moreno); 000000023326557X (D. Jimenez); 0009-0007-7198-6849 (L. Moreno-Mendieta)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

for automatic content moderation systems. The same terms can express hate or, conversely, pride and community, depending on the discursive context, communicative intent, and speaker identity. This semantic ambiguity makes traditional approaches to hate speech detection, based on simplistic binary classifications, insufficient and potentially harmful, as they may censor legitimate expressions of the LGBTQ+ community. In recent years, Natural Language Processing (NLP) has shown notable progress in the automatic detection of hate speech, particularly through models based on transformer architectures such as mBERT [1] and XLM-R [2], which have demonstrated strong performance in multilingual and multicultural settings [3, 4]. However, most of these approaches tend to focus on binary hate classification and often overlook alternative, ironic, or identity-based uses of traditionally pejorative terms, which can lead to bias and censorship of legitimate LGBTQ+ discourse.

At an international level, competitions such as LT-EDI (Language Technology for Equality, Diversity, and Inclusion) have fostered the development of NLP approaches for classifying homophobia and transphobia in social media comments, demonstrating that transformer-based models significantly outperform traditional methods in multilingual tasks [3, 4]. In the Spanish-speaking world, the HOMO-MEX competition organized by IberLEF (Iberian Languages Evaluation Forum) in 2023 provided the first annotated corpus of Mexican Spanish tweets for the detection of LGBTQ+-phobia. By 2024, the challenge expanded into more complex domains such as song lyrics. These efforts have shown that while Spanish-language transformers such as BETO [5] and multilingual models like mBERT [1] can achieve promising results, they also revealed the difficulty of capturing implicit forms of hate as well as reclamation phenomena and dialectal variation typical of Latin American sociocultural contexts.

In this context, MultiPRIDE [6] (Multilingual Automatic Detection of Reclamation of Slurs in the LGBTQ+ Context) emerges as a shared task within EVALITA 2026, specifically focused on the reclamation of offensive terms toward the LGBTQ+ community across different languages and linguistic varieties. Unlike previous tasks focused exclusively on hate speech detection, MultiPRIDE addresses the semantic ambiguity of historically stigmatizing expressions, distinguishing between discriminatory uses and reclaimed uses for identity, humor, or empowerment purposes. It is a multilingual binary classification task involving data in Italian, Spanish, and English, with the goal of determining whether a term related to the LGBTQ+ context in a sentence is being used with reclaiming intent or not.

This task seeks to provide resources and models that enable fairer and more contextualized content moderation, capable of protecting LGBTQ+ individuals from hate speech without silencing their own forms of expression and self-affirmation. MultiPRIDE invites researchers, professionals, and students to explore the linguistic features and challenges associated with reclaimed language within the LGBTQ+ community, considering both textual content (arguments, slurs, derogatory words, self-labeling, rhetorical figures) and contextual information that can be inferred from user profiles, such as their LGBTQ+ identity.

In this paper, we propose the design and implementation of two deep learning models to address the MultiPRIDE task, incorporating preprocessing and representation strategies tailored to Spanish and its regional varieties. [Placeholder for results and analysis]. Our goal is to advance toward automatic moderation systems that not only detect hate speech but also recognize and respect the linguistic reclamation processes that characterize the diversity and vitality of LGBTQ+ discourse online.

2. State of the Art

The problem of linguistic reclamation in the LGBTQ+ context is closely linked to recent developments in automatic hate speech detection research. Although NLP models based on transformers have demonstrated remarkable capacity to identify explicit expressions of homophobia and transphobia, the literature has consistently shown that these systems present substantial limitations when confronting

ironic, community-based, or reclaimed uses of historically offensive terms [7, 8]. In early work on automatic detection of homophobia and transphobia, such as those promoted by LT-EDI [9, 3, 10, 4], the main objective was to distinguish between offensive and non-offensive content across multiple languages. While these efforts consolidated the use of architectures such as mBERT [1] and XLM-R [2] as the de facto standard for hate detection, the annotation and evaluation schemes were primarily based on binary or multiclass classifications that do not explicitly model the semantic ambiguity of stigmatizing terms. In parallel, resources such as HOMO-MEX [11, 12, 13] revealed that, even in Spanish, the performance of models like BETO [5] and mBERT [1] degrades significantly when discourse contains irony, cultural implicatures, or community-based reappropriations, as frequently occurs in creative domains such as song lyrics. These findings suggest that a significant portion of classification error stems not from the model’s inability to detect offensive terms, but from its difficulty interpreting their pragmatic function within discourse [14, 15].

This problem has been addressed more explicitly in recent work on queer language and model biases. Andersen et al. [16] demonstrated that the semantic polarity of terms related to the LGBTQ+ community in Mexican Spanish has changed significantly over time, meaning that models trained on historical data tend to overestimate the offensive character of words that have been partially reclaimed. Complementarily, recent studies have shown that language models tend to label discourse produced by queer speakers as harmful more frequently than heteronormative discourse, even when it contains no explicit attacks, revealing a systematic bias against queer dialect and identity-based self-labeling [17, 18, 19]. This phenomenon extends beyond LGBTQ+ contexts, as similar biases have been documented for African American English and other marginalized linguistic varieties [15, 20].

In the Italian context, the ReCLAIM project [21] has directly addressed the phenomenon of slur reappropriation through the construction of resources and models designed to distinguish between discriminatory and identity-based uses of stigmatizing terms, showing that contextual and discursive information is essential for reliable classification. These results reinforce the need to move from purely lexical or sentiment-based hate detection schemes toward models capable of capturing pragmatic relationships, community affiliation, and discursive positioning [22, 23]. The psychological and sociolinguistic dimensions of slur reclamation have been well documented. Galinsky et al. [22] established that group power mediates the willingness to self-label with derogatory terms, and that such self-labeling can increase both individual and collective perceptions of power while attenuating the stigma associated with these terms. From a linguistic perspective, Anderson and Lepore [24] analyzed the semantic mechanisms through which slurs operate, providing theoretical foundations for understanding how the same lexical items can function differently depending on speaker identity and communicative context. In this framework, MultiPRIDE represents a natural evolution of previous LGBTQ+-phobia detection tasks by shifting the focus from mere hate identification toward explicit modeling of the semantic and pragmatic ambiguity associated with linguistic reclamation processes. By formulating the task as a binary classification between reclaimed and non-reclaimed uses of LGBTQ+ terms across multiple languages (Italian, Spanish, and English), MultiPRIDE directly connects with the findings of LT-EDI, HOMO-MEX, and ReCLAIM, and provides an experimental framework for evaluating the extent to which current models can distinguish between symbolic violence and discursive self-affirmation.

From this perspective, the design of the models proposed in this work is grounded in the accumulated evidence that hate speech detection toward the LGBTQ+ community cannot be reduced to the presence of offensive lexicon, but rather requires incorporating signals of context, identity, and pragmatic function [25]. In this way, our proposal seeks to contribute to the development of automatic moderation systems that, in line with MultiPRIDE’s objectives, protect LGBTQ+ individuals from linguistic harm without rendering invisible or penalizing their own forms of expression, resistance, and linguistic reappropriation [26].

3. Methodology

A brief description of the provided datasets for the competition is given in this section; furthermore, a detailed description of the method selected to resolve the problem is provided.

3.1. Task Description

The MultiPride 2026 shared task focuses on the classification of reclaimed language within the LGBTQ+ community on social media. The task invites participants to explore linguistic features and contextual elements associated with language reclamation, a phenomenon where marginalized communities reappropriate terms that were originally used as slurs or derogatory labels.

Participants are encouraged to analyze both the textual content of the inputs—including arguments, slurs, derogatory words, self-labeling practices, and figures of speech—and the contextual information that can be inferred from users’ profiles when available, such as community membership and political orientation.

The primary objective is to perform binary classification, determining whether a term related to the LGBTQ+ context within a given sentence is used with reclamatory intent or not. The shared task is organized into two main tasks:

3.1.1. Task A - Textual Content

In Task A, participants are provided exclusively with the textual content of messages. The task can be approached in two distinct modalities:

- **Constrained approach:** Participants must utilize only the provided training data. While additional resources such as lexicons are permitted, the use of supplementary training data in the form of tweets or sentences is prohibited.
- **Unconstrained approach:** Participants may leverage additional training data, including external datasets annotated for reclaimed language. Systems developed using this approach must be clearly identified during run submission and thoroughly detailed in the technical report.

Task A comprises three language-specific subtasks. Participants may choose to work on one, two, or all three languages, with multilingual experiments being particularly encouraged:

- **Subtask A1 - Italian:** Classification of reclaimed language in Italian texts.
- **Subtask A2 - Spanish:** Classification of reclaimed language in Spanish texts.
- **Subtask A3 - English:** Classification of reclaimed language in English texts.

3.1.2. Task B - Contextual Content

Task B extends beyond textual analysis by incorporating contextual information derived from authors’ profiles, including biographical data when available. This task is organized into two language-specific subtasks:

- **Subtask B1 - Italian:** Classification leveraging both Italian texts and user profile information.
- **Subtask B2 - Spanish:** Classification leveraging both Spanish texts and user profile information.

It should be noted that biographical information is available exclusively for Spanish and Italian datasets. As with Task A, participants may choose to work on one or both languages, and cross-linguistic analysis is strongly encouraged.

3.1.3. Dataset Examples

The following table shows some examples of samples extracted from the three available languages. It should be noted that the English language samples do not have a user or biography, which prevents task 2 from existing for this language.

Table 1

Reclaimed and not-reclaimed data examples from the multilingual dataset.

Lang	User	Bio	Tweet	Reclamation
it	@user	Certo le circostanze non sono favorevoli.	In quanto disabile e frocia questi sono i miei Pride-Months. Ma vorrei anche dire che il giorno in cui nel manifesto di un evento lebtqiatransfeminista verrà citato anche l’antiabilismo oltre ad anti sessismo/obitfobia/razzismo/specismo offro da bere	yes
it	@user	I veri partigiani furono i primi sovranisti! w la patria!	Ecco, adesso pensate all’iter o in affitto ed al male che fate al bambino branco di finocchi arcobaleno	no
es	@user	Me llaman feminista, roja y bollera.	Buenas tardes a rojos, feministas, republicanos, maricones, bolleras y demás LGTBI #LGTBI #pride	yes
es	@user	I live for that energy!	Hace rato pasó una caravana de movimiento LGT...etc y algunos me vieron parado observando y me gritaron que yo también era marica. Órale, bien «respetuosas» estas personas que exigen respeto. #PrideMonth #Pride2022 #LGTBQ	no
es	N/A	N/A	I use the word tranny all the time...but that’s only in reference to working on my cars. Transgendered, transvestite, and drag queen folk are too fabulous to have their descriptions abbreviated.	yes
es	N/A	N/A	Actually that’s what s faggot is. Fag is just something that needs to be burnt.	no

3.2. Corpus Description

The MultiPride 2026 shared task provides a unified training set for both Task A (Textual Content) and Task B (Contextual Content). The training data is distributed across three language-specific CSV files, each corresponding to one of the supported languages:

- `train_it.csv` – Italian language dataset
- `train_en.csv` – English language dataset
- `train_es.csv` – Spanish language dataset

3.2.1. Data Format

Each training file follows a standardized format consisting of five fields, as described in Table 2:

The `id` field serves as a unique identifier for each message in the dataset, enabling traceability and reproducibility of experimental results. The `text` field contains the actual message content extracted from social media posts. The `label` field represents the gold standard annotation, indicating whether the text contains terms used with reclamatory intent.

The `bio` field provides contextual information about the message author through their profile biography. This field is particularly relevant for Task B, where contextual information is leveraged for classification. It is important to note that biographical data is not available for all users; consequently, this field may contain missing values. Additionally, biographical information is available exclusively for Italian and Spanish datasets, while the English dataset does not include this feature.

Finally, the `lang` field specifies the primary language of each message, facilitating language-specific processing and enabling multilingual analysis across the three supported languages.

Table 2
Training data format specification.

Field	Description
id	Unique identifier for each message
text	The textual content of the message
label	Binary annotation indicating whether the text contains reclaimed language (reclamatory intent)
bio	User profile biography (when available); this field may contain null values for messages where biographical information is unavailable
lang	Primary language of the text (it, en, or es)

3.3. Data Loading and Preparation

The training and test datasets were stored in Amazon S3 cloud storage and accessed programmatically using the AWS SDK for Python (Boto3). A dedicated function was implemented to retrieve CSV files from specified S3 buckets and prefixes, handling potential errors during the data retrieval process.

The data loading procedure consisted of two main phases. First, the training data was loaded from three separate language-specific files: `train_es.csv`, `train_en.csv`, and `train_it.csv`, corresponding to Spanish, English, and Italian datasets, respectively. Upon successful retrieval of each file, the number of rows was verified to ensure data integrity. The three training datasets were subsequently concatenated into a unified dataframe, facilitating multilingual model training and cross-linguistic analysis.

Following the consolidation of training data, descriptive statistics were computed, including the total number of training samples and the distribution of labels across the binary classification task. This analysis provided insights into the class balance within the training corpus, which is crucial for understanding potential biases and for implementing appropriate data handling strategies during model development.

In the second phase, the test datasets were retrieved following an analogous procedure. Three language-specific test files—`es_test.csv`, `en_test.csv`, and `it_test.csv`—were loaded from their designated S3 location. These test sets were maintained separately from the training data to enable proper evaluation of the developed models on unseen data, ensuring unbiased performance assessment across all three languages.

3.4. Knowledge Graph Construction

To effectively capture the complex semantic relationships and contextual nuances associated with reclaimed language in the LGBTQ+ community, a Knowledge Graph (KG) was constructed using the NetworkX library for Python. The KG serves as a structured repository of linguistic knowledge, encoding information about reclaimed terms, their usage contexts, and associated semantic attributes across the three target languages.

3.4.1. Graph Structure and Initialization

The Knowledge Graph was implemented as a directed graph, where nodes represent either reclaimed terms or contextual categories, and edges encode relationships between these entities. The graph structure was designed to facilitate efficient querying and reasoning about term usage patterns and contextual appropriateness.

Upon initialization, the KG was populated with a curated set of known reclaimed terms across all three languages. Each term node was annotated with language-specific metadata, including:

- **Language identifier:** Specifying whether the term belongs to Spanish (es), English (en), or Italian (it)

- **Term type:** Classifying the node as a reclaimed term or context category
- **Context attributes:** Providing information about the typical usage context and semantic connotations of the term

3.4.2. Multilingual Term Repository

The KG incorporates language-specific reclaimed terms based on sociolinguistic literature and community usage patterns:

Spanish terms: The Spanish lexicon includes terms such as *marica*, *maricón*, *bollera*, *tortillera*, *joto*, and *jota*. These terms were annotated with their predominant contexts, ranging from positive LGBTQ+ usage to heavily context-dependent interpretations. Regional variations were also considered, particularly for terms like *joto* and *jota*, which are predominantly used in Mexican Spanish contexts.

English terms: The English repository encompasses widely recognized reclaimed terms including *queer*, *dyke*, *fag*, *faggot*, and *trannie*. Each term was classified according to its reclamation status within the LGBTQ+ community, with particular attention to terms that remain controversial or whose appropriateness varies significantly by context and speaker identity.

Italian terms: The Italian lexicon includes terms such as *frocio*, *checca*, *ricchione*, and *finocchio*. Similar to the other languages, these terms were annotated with contextual information reflecting their complex usage patterns within Italian LGBTQ+ communities.

3.4.3. Context Nodes and Semantic Categories

In addition to term nodes, the KG incorporates abstract context nodes representing different usage environments. These include:

- *lgbtq_community*: Indicating usage within community spaces
- *pride_context*: Associated with pride events and celebrations
- *activist_context*: Related to advocacy and activism
- *derogatory_context*: Indicating pejorative usage
- *neutral_context*: Representing ambiguous or neutral usage

These context nodes enable the encoding of relationships between terms and their typical usage environments, facilitating context-aware classification.

3.4.4. Term Extraction and Matching

The KG provides functionality for identifying potential reclaimed terms within input texts. The extraction mechanism operates through pattern matching, comparing normalized (lowercased) text against the terms stored in the graph. Language-specific filtering ensures that only terms corresponding to the declared language of each message are considered as candidates, preventing false matches from cross-linguistic homographs or similar forms.

3.4.5. Contextual Analysis Framework

A key component of the Knowledge Graph system is its contextual analysis capability, which evaluates the semantic environment surrounding identified terms. This analysis leverages both the textual content and available biographical information to assess the likelihood of reclamatory usage.

The contextual analysis framework employs two sets of linguistic indicators:

Positive indicators: Terms and expressions associated with affirmative LGBTQ+ discourse were identified, including explicit community markers (*pride, orgullo, LGBTQ+, comunidad, community*), activist terminology (*activism, visibility, visibilidad*), and values-oriented language (*love, amor, equality, igualdad, diversity, diversidad, respect, respeto*). Additionally, symbolic representations such as rainbow flag emojis and heart emojis were included as positive indicators.

Negative indicators: Conversely, terms associated with hostile or discriminatory discourse were catalogued, including explicit hate speech markers (*hate, odio*), oppositional language (*against, contra*), and specific forms of prejudice (*homophobia, homofobia, transphobia, transfobia, discrimination, discriminación*).

For each input message, the contextual analysis function computes the frequency of positive and negative indicators in both the message text and the author’s biographical information when available. The combined frequency counts are used to derive a sentiment classification (positive, negative, or neutral) and to assess the presence of LGBTQ+-affirming context. This information provides valuable features for downstream classification models.

3.4.6. Term Information Retrieval

The KG also provides a query interface for retrieving detailed information about specific terms. This functionality allows models to access the stored attributes and contextual annotations for any term present in the graph, enabling informed decision-making during the classification process.

The resulting Knowledge Graph comprises nodes representing both linguistic terms and semantic contexts, providing a structured knowledge base that encodes expert understanding of reclaimed language phenomena across the three target languages.

3.5. Knowledge Graph Enhancement from Training Data

While the initial Knowledge Graph was populated with expert-curated reclaimed terms based on sociolinguistic literature, the diversity and evolution of language use within LGBTQ+ communities necessitates data-driven enrichment. To this end, a learning mechanism was implemented to extract additional patterns and contextual information from the labeled training data, thereby augmenting the KG with empirically observed usage patterns.

3.5.1. Training Data Analysis Strategy

The enhancement process focused exclusively on positive examples—that is, messages labeled as containing reclaimed language (label = 1). This selective approach was motivated by the objective of identifying authentic reclamation patterns as manifested in actual community discourse. By analyzing confirmed instances of language reclamation, the system could capture genuine usage contexts, associated linguistic features, and co-occurring elements that characterize reclamatory intent.

The analysis was conducted in a language-stratified manner, processing Spanish, English, and Italian examples independently. This language-specific approach ensured that the extracted patterns and contextual associations remained linguistically and culturally appropriate, respecting the distinct reclamation dynamics that exist across different linguistic communities.

3.5.2. Feature Extraction from Reclaimed Examples

For each reclaimed example in the training data, multiple levels of linguistic analysis were performed to extract salient features:

Hashtag extraction: Social media hashtags serve as important contextual markers, often explicitly signaling thematic content, community affiliation, and communicative intent. A pattern matching mechanism was employed to identify and extract all hashtags present in reclaimed examples. These

hashtags were normalized to lowercase to facilitate consistent matching and comparison. Common hashtags in reclaimed language contexts include pride-related tags (e.g., *#PrideMonth*, *#Pride2022*), community identifiers (e.g., *#LGBTQ*, *#LGTBI*), and specific event or movement markers.

Term identification: For each training example, the previously described term extraction functionality was applied to identify which reclaimed terms from the KG appeared in the text. This process leveraged the language-specific term repositories established during initialization, ensuring accurate identification of relevant lexical items.

Contextual association: When reclaimed terms were identified in a training example, the system created an association between the term and its observed usage context. For each occurrence, the following information was recorded:

- A textual excerpt (limited to 100 characters) providing immediate context around the term usage
- The binary label confirming reclamatory intent
- The set of hashtags co-occurring in the same message

This contextual information was stored in a term-specific repository, creating a corpus of authentic usage examples for each reclaimed term in the KG.

3.6. Vector Database and Retrieval-Augmented Generation System

To leverage the rich contextual information present in the training data, a Retrieval-Augmented Generation (RAG) system was implemented. This system enables the classification model to retrieve and reference similar examples from the training corpus when making predictions on new instances, effectively implementing a case-based reasoning approach informed by empirically observed usage patterns.

3.6.1. Embedding Model Selection

The foundation of the RAG system is a multilingual sentence embedding model, specifically the *paraphrase-multilingual-MiniLM-L12-v2*¹ model from the Sentence Transformers library. This model was selected for its several advantageous properties:

- **Multilingual coverage:** The model supports over 50 languages, including Spanish, English, and Italian, enabling consistent semantic representation across all three target languages of the shared task.
- **Semantic preservation:** The model is trained to generate embeddings that capture semantic similarity rather than mere lexical overlap, allowing the system to identify contextually similar examples even when surface forms differ.
- **Efficiency:** As a distilled variant (MiniLM), the model offers a favorable balance between representation quality and computational efficiency, facilitating real-time retrieval during inference.
- **Cross-lingual alignment:** The model's training on multilingual parallel corpora enables meaningful similarity comparisons across languages, which is particularly valuable for cross-linguistic analysis and transfer learning scenarios.

3.6.2. System Architecture

The RAG system comprises three primary components: an embedding encoder, a vector index for efficient similarity search, and a repository of training examples with their associated labels. The embedding encoder transforms textual inputs into dense vector representations in a high-dimensional

¹<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>

semantic space. The vector index, implemented using the FAISS² (Facebook AI Similarity Search) library, enables efficient nearest-neighbor search over large collections of embeddings. The example repository maintains the original training instances and their labels, allowing the system to return complete contextual information for retrieved examples.

3.6.3. Vector Index Construction

The construction of the vector database follows a multi-stage pipeline designed to maximize the informational content of embeddings while maintaining computational tractability.

Text preprocessing and augmentation: For each training example, a composite representation was constructed by combining the message text with available biographical information. Specifically, when user biography was available (i.e., for Italian and Spanish examples in Task B), the text was augmented with the biographical content using a structured format: the message text followed by a [BIO: . . .] marker containing the biography. This augmentation strategy ensures that the resulting embeddings capture both the immediate textual context and the broader author-level context, which is particularly relevant for distinguishing reclamatory from derogatory usage.

For Task A scenarios, where contextual information is excluded, only the message text is used to generate embeddings, ensuring that the retrieval mechanism respects the task constraints.

Batch embedding generation: The composite texts were processed through the embedding model in batches of 32 examples, balancing memory efficiency with processing speed. The model generates fixed-dimensional dense vector representations (embedding dimension determined by the model architecture) that encode semantic content in a continuous space where semantically similar texts are positioned proximally.

The batch processing approach with progress tracking ensures transparency during the potentially time-consuming embedding generation phase, particularly important when processing large training corpora.

FAISS index construction: The generated embeddings were used to construct a FAISS index employing the L2 (Euclidean) distance metric. Specifically, an `IndexFlatL2` index was utilized, which performs exact nearest-neighbor search without approximation. While approximate methods (e.g., `IndexIVFFlat`) could offer computational advantages for very large datasets, the exact search approach was deemed appropriate given the dataset size and the importance of retrieval accuracy for classification performance.

The embeddings were cast to 32-bit floating-point format before index insertion, adhering to FAISS's memory layout requirements while maintaining sufficient numerical precision for similarity computations.

Metadata preservation: Alongside the vector index, the system maintains parallel data structures storing the complete training examples (as dictionary records) and their corresponding labels. This design enables the retrieval mechanism to return not only similar embeddings but also the full contextual information associated with retrieved examples, including text, biography, language, and label information.

3.6.4. Retrieval Mechanism

The retrieval system accepts a query consisting of a text message and optional biographical information, and returns the k most similar training examples based on semantic similarity in the embedding space.

²<https://ai.meta.com/tools/faiss/>

Query processing: Query texts undergo the same preprocessing and augmentation steps applied during index construction. For Task B (contextual content), queries are augmented with biographical information when available, ensuring consistency between query and index representations. For Task A (textual content only), queries consist solely of the message text, excluding biographical context.

The query text is encoded using the same embedding model employed during index construction, generating a query embedding in the same semantic space as the indexed training examples.

Similarity search: The FAISS index performs efficient nearest-neighbor search, identifying the training examples whose embeddings exhibit minimal L2 distance to the query embedding. The search can be configured to retrieve a specified number of neighbors (k), allowing flexible control over the breadth of retrieved context.

Language-specific filtering: To respect language boundaries and prevent cross-linguistic interference, the retrieval mechanism supports optional language filtering. When a target language is specified, the system initially retrieves a larger candidate set (three times the desired k value) and then filters to retain only examples matching the target language. This over-retrieval and filtering strategy ensures that the final result set contains the desired number of same-language examples, which is particularly important for language-specific subtasks.

Similarity score computation: The L2 distances returned by FAISS are transformed into normalized similarity scores using the formula $s = 1/(1 + d)$, where d represents the L2 distance. This transformation maps distances to a $[0, 1]$ interval where higher values indicate greater similarity, providing an intuitive measure of example relevance.

3.6.5. Task-Specific Retrieval Strategies

The RAG system implements distinct retrieval strategies for Task A and Task B, reflecting the different information modalities available in each task:

- **Task A (Textual Content):** Retrieval is based exclusively on message text, with biographical information explicitly excluded from both query and index representations. This ensures that the system learns to identify reclamation based solely on linguistic content.
- **Task B (Contextual Content):** Retrieval incorporates both textual and biographical information, enabling the system to identify similar examples based on both what is said and who says it. This richer representation allows the model to leverage author-level context for disambiguation.

The resulting vector database provides a comprehensive semantic index of the training corpus, enabling efficient retrieval of relevant examples that inform classification decisions through case-based reasoning.

3.7. Large Language Model Integration

The classification system leverages Claude 4.5 Haiku, a state-of-the-art large language model developed by Anthropic, as the primary decision-making component. The model is accessed through Amazon Bedrock, a fully managed service that provides API access to foundation models. Specifically, the model identifier `us.anthropic.claude-3-5-haiku-20241022-v1:0` was utilized, representing the Claude 3.5 Haiku variant optimized for efficient inference while maintaining strong reasoning capabilities.

3.7.1. Classifier Architecture

The classifier implements a hybrid architecture that integrates three complementary knowledge sources: expert-encoded linguistic knowledge (Knowledge Graph), empirical usage patterns (RAG system), and the general linguistic and cultural understanding encoded in Claude’s parameters. This multi-source approach enables the system to leverage structured knowledge, case-based reasoning, and large-scale language understanding in a unified framework.

The classifier maintains references to both the Knowledge Graph and RAG system, enabling dynamic context retrieval during inference. Additionally, it establishes a connection to AWS Bedrock Runtime, which handles the low-level details of API communication, request formatting, and response parsing.

3.7.2. AWS Bedrock Integration

AWS Bedrock provides a standardized interface for accessing multiple foundation models, including the Claude family of models. The integration requires proper AWS credentials and IAM permissions, specifically the `bedrock:InvokeModel` permission. The classifier instantiates a Bedrock Runtime client configured for the appropriate AWS region, enabling programmatic invocation of Claude models.

The Bedrock API follows the Messages format specified by Anthropic, requiring structured request bodies that include:

- **API version identifier:** Specifying the Bedrock-specific Anthropic API version (`bedrock-2023-05-31`)
- **Token budget:** Maximum number of tokens Claude may generate in its response (set to 1000 to accommodate structured JSON output with reasoning)
- **Message history:** An array of conversational turns, in this case containing a single user message with the classification prompt

Model invocation returns a response object containing the generated text, which is subsequently parsed to extract classification decisions.

3.7.3. Prompt Engineering Strategy

A critical component of the system is the prompt construction mechanism, which transforms raw input data and retrieved context into a comprehensive instruction for Claude. The prompt engineering strategy employs several techniques to optimize classification performance:

Task specification and definitions: The prompt begins with explicit role assignment, establishing Claude as an expert in LGBTQ+ linguistics and reclaimed language analysis. This priming encourages the model to activate relevant knowledge and adopt an appropriate analytical perspective.

Crucially, the prompt provides precise definitions of key concepts, particularly the distinction between reclaimed language (label 1) and other forms of LGBTQ+-related discourse (label 0). The definition emphasizes that reclamation requires not merely the presence of potentially reclaimed terms, but their use in a positive, affirming, or community-building context. This definitional clarity helps address a common source of classification errors: the conflation of neutral informational content about LGBTQ+ topics with authentic language reclamation.

Contextual principles: The prompt articulates key principles for classification, emphasizing that:

1. Context is paramount—the same lexical item can function as reclaimed or derogatory depending on usage
2. Community membership indicators provide crucial disambiguation signals
3. Author biographical information, when available, offers valuable context
4. Mere mention of LGBTQ+ topics or terminology does not constitute reclamation

These principles encode sociolinguistic insights about language reclamation phenomena, guiding Claude toward contextually appropriate interpretations.

Knowledge Graph context injection: When the Knowledge Graph identifies potential reclaimed terms in the input text, this information is incorporated into the prompt. Specifically, the prompt includes:

- The list of detected terms from the KG’s multilingual lexicon
- A binary indicator of whether LGBTQ+-affirming context markers were identified
- Quantified counts of positive and negative sentiment indicators

This structured information provides Claude with explicit signals about the linguistic content and semantic environment of the message, complementing its own pattern recognition capabilities.

Few-shot learning via retrieved examples: The prompt incorporates a few-shot learning component by including the top-3 most similar training examples retrieved by the RAG system. For each retrieved example, the prompt presents:

- The message text (truncated to 150 characters to manage token budget)
- The gold-standard label, explicitly indicating whether the example represents reclaimed or non-reclaimed usage

This few-shot approach enables in-context learning, allowing Claude to observe concrete instances of classification decisions and adapt its reasoning accordingly. The retrieved examples are selected based on semantic similarity to the test instance, ensuring that the provided examples are contextually relevant.

Test instance presentation: The message to be classified is clearly demarcated within the prompt, along with associated metadata:

- The complete message text
- Author biographical information (for Task B only, when available)
- Language identifier (Spanish, English, or Italian)

This structured presentation ensures Claude can clearly distinguish between contextual information and the actual classification target.

Output format specification: To facilitate automated parsing of Claude’s responses, the prompt explicitly requests structured JSON output with three fields:

- `label`: Binary classification decision (0 or 1)
- `confidence`: Numerical confidence score on a 0-1 scale
- `reasoning`: Brief textual explanation of the classification decision

The prompt explicitly instructs Claude to respond with JSON only, without additional markdown formatting or commentary, ensuring parseable outputs. The inclusion of reasoning provides interpretability, enabling qualitative analysis of the model’s decision-making process.

3.7.4. Classification Process

For each input instance, the classification process follows a multi-stage pipeline:

Context retrieval phase: The system first queries the Knowledge Graph to identify potential re-claimed terms and analyze contextual indicators. Concurrently, the RAG system retrieves semantically similar training examples. These operations execute independently and can be parallelized if needed.

Prompt construction phase: The retrieved contexts are integrated with the input instance to construct a comprehensive prompt following the strategy described above. Language-specific information ensures appropriate linguistic framing.

Model invocation phase: The constructed prompt is formatted into a Bedrock API request and transmitted to Claude 4.5 Haiku. The model processes the prompt and generates a response containing the classification decision.

Response parsing phase: The system extracts the JSON object from Claude's response using pattern matching. A regular expression identifies JSON structures containing the required `label` field, which is then parsed to extract the classification decision, confidence score, and reasoning.

3.7.5. Task-Specific Processing

The classifier supports both Task A (textual content only) and Task B (textual and contextual content) through a configurable parameter. When processing Task A instances, biographical information is excluded from both retrieval (in the RAG system) and prompt construction, ensuring that decisions are based solely on textual content. For Task B, biographical information is fully integrated when available.

3.7.6. Batch Processing

For efficient evaluation on test sets, the classifier implements a batch processing interface that iterates over all instances in a dataset, applies the classification pipeline to each instance, and aggregates results. Progress tracking via the `tqdm` library provides real-time feedback during long-running inference processes.

The batch processor returns a structured dataframe containing classification results, confidence scores, reasoning, and metadata for each instance. This output format facilitates subsequent analysis, error diagnosis, and submission formatting.

3.7.7. Error Handling and Robustness

The system implements comprehensive error handling to maintain operational stability:

- **API errors:** Network failures, timeouts, and service errors are caught and trigger fallback classification strategies
- **Permission errors:** Initialization checks verify Bedrock access permissions and provide diagnostic messages if permissions are insufficient
- **Parsing failures:** Malformed responses are handled gracefully through multiple parsing strategies of increasing leniency
- **Rate limiting:** While not explicitly implemented, the sequential processing approach naturally respects API rate limits

This robust error handling ensures that the system can complete evaluation even when encountering occasional failures, while logging issues for subsequent investigation.

3.8. System Validation on Training Sample

Prior to evaluation on the official test sets, the classification system underwent validation on a held-out sample from the training data. This validation phase served multiple purposes: verifying the operational integrity of the complete pipeline, assessing the system’s performance characteristics, and comparing the effectiveness of Task A (textual content only) versus Task B (textual and contextual content) approaches.

3.8.1. Validation Set Construction

A validation subset of 30 instances was randomly sampled from the complete training corpus using stratified random sampling with a fixed random seed (seed = 42) to ensure reproducibility. The relatively small validation set size was deliberately chosen to enable rapid iteration and system debugging while managing computational costs associated with API calls to the Claude model.

The sampling process maintained the natural distribution of labels within the training data, providing a representative subset that reflects the class balance characteristics of the full corpus. The label distribution within the validation set was explicitly recorded to facilitate interpretation of validation results and to ensure awareness of any class imbalance effects.

3.8.2. Dual-Task Evaluation Framework

The validation process evaluated the system under both task configurations to quantify the added value of biographical context:

Task A validation (textual content only): The classifier processed each validation instance using only the message text, explicitly excluding biographical information from both the RAG retrieval process and the prompt construction. This configuration tests the system’s ability to identify reclaimed language based purely on linguistic content, representing the most challenging scenario where contextual disambiguation must rely solely on textual cues.

The classification pipeline generated predictions for all 30 validation instances, producing structured outputs containing predicted labels, confidence scores, and reasoning explanations. The predicted labels were then compared against the gold-standard annotations to compute validation accuracy.

Task B validation (textual and contextual content): The same validation instances were re-processed with biographical information included. For instances where user biography was available, this information was incorporated into both the similarity-based retrieval (enabling identification of contextually similar examples) and the prompt (providing Claude with explicit author context).

This configuration represents the full-information scenario where both linguistic content and author identity markers contribute to classification decisions. The comparison between Task A and Task B performance quantifies the informational value of biographical context for reclaimed language identification.

3.8.3. Performance Metrics and Analysis

For each task configuration, validation accuracy was computed as the proportion of correctly classified instances. Additionally, the predicted label distribution was analyzed to identify any systematic biases toward particular classes (e.g., over-prediction of reclamation or non-reclamation).

The merging of predictions with ground-truth labels facilitated instance-level error analysis, enabling identification of challenging cases where the system failed to produce correct classifications. This granular view supports understanding of system strengths and weaknesses.

3.8.4. Qualitative Analysis

Beyond aggregate accuracy metrics, a qualitative analysis examined individual predictions to assess the interpretability and reasoning quality of the system. For a subset of validation instances (typically 3-5 examples), the following information was reviewed:

- **Input text:** The message content being classified (truncated for display purposes)
- **Ground truth label:** The gold-standard annotation from the training data
- **Predicted label:** The system's classification decision
- **Confidence score:** The system's self-assessed certainty (0-1 scale)
- **Reasoning:** Claude's explanation for its classification decision

This qualitative review served multiple analytical purposes. First, it provided insight into the system's decision-making process, revealing which textual and contextual features influenced classifications. Second, it enabled identification of reasoning patterns that distinguish correct from incorrect predictions. Third, it helped diagnose potential failure modes, such as over-reliance on superficial lexical cues or insufficient sensitivity to contextual nuance.

The reasoning text generated by Claude proved particularly valuable for system refinement, as it made explicit the factors considered during classification. For instance, correct predictions often cited specific indicators of community membership, positive framing, or reclamatory intent, while incorrect predictions sometimes revealed misinterpretations of ambiguous language or insufficient weight given to biographical context.

Here are some examples of quantitative analysis predictions where you can see why the label is being assigned.

1. **Text:** "Quit being a faggot and suck that dick" Oh my I lost it ...
True: 1, **Predicted:** 0
Confidence: 0.90
Reasoning: Quote contains a derogatory slur used in a homophobic context, not a positive reclamation by LGBTQ+ ...
2. **Text:** Un lujo colaborar desde @USER con @USER en la semana de Cine Queer en el marco del mes de la diversi...
True: 0, **Predicted:** 0
Confidence: 0.85
Reasoning: While the tweet contains 'Queer', it is used in an informational context about a film event, not as ...
3. **Text:** come quando "ma frocio non mi sembra un insulto" amore mio se io parte della comunità lgbtqia ti dic...
True: 0, **Predicted:** 0
Confidence: 0.90
Reasoning: The tweet discusses the term 'frocio' as an offensive slur, critiquing someone who claims it's not h...

3.8.5. Comparative Performance Assessment

A key objective of the validation phase was quantifying the performance differential between Task A and Task B configurations. The accuracy difference (Task B accuracy minus Task A accuracy) provides a direct measure of how much biographical information contributes to classification performance on this validation sample.

Several scenarios were anticipated:

1. **Positive differential:** Task B outperforming Task A would indicate that biographical context provides disambiguating information that improves classification, validating the inclusion of this feature in the full system.

2. **Zero differential:** Equal performance would suggest that biographical information, at least in this validation sample, does not substantially aid classification, possibly because textual content alone provides sufficient signal.
3. **Negative differential:** Task A outperforming Task B (an unlikely but possible outcome) would suggest that biographical information introduces noise or causes overfitting to spurious correlations.

The observed performance differential, along with analysis of prediction distributions and individual cases, informed final system configuration decisions and provided insights into the relative importance of different information modalities for reclaimed language classification.

3.8.6. Validation Insights and System Refinement

The validation phase provided actionable insights that guided system refinement:

- **Prompt effectiveness:** Analysis of reasoning patterns revealed which prompt components (definitions, principles, retrieved examples, KG context) most strongly influenced correct decisions, informing prompt optimization.
- **RAG system performance:** Examination of retrieved examples for misclassified instances helped assess whether retrieval was surfacing relevant precedents or introducing misleading analogies.
- **Knowledge Graph coverage:** Validation errors sometimes revealed reclaimed terms or usage patterns not captured in the KG, suggesting areas for knowledge base expansion.
- **Confidence calibration:** Comparison of confidence scores with classification accuracy enabled assessment of whether the system’s self-reported certainty correlates with actual correctness.

4. Experiments and Results

This section presents the evaluation of the proposed system on the MultiPride 2026 test sets across all subtasks and languages. The experimental design encompasses both Task A (textual content only) and Task B (textual and contextual content), enabling comprehensive assessment of the system’s performance and the contribution of biographical information to classification accuracy.

4.1. Experimental Setup

The evaluation was conducted on the official test sets provided by the shared task organizers. Three language-specific test sets were available:

- **Spanish:** 585 test instances for both Task A and Task B
- **English:** 685 test instances for Task A only (biographical information not available)
- **Italian:** 725 test instances for both Task A and Task B

For each test instance, the system generated binary classification predictions (0 for not reclaimed, 1 for reclaimed) along with confidence scores ranging from 0.0 to 1.0. All predictions were obtained using the complete pipeline described in previous sections, incorporating Knowledge Graph analysis, RAG-based example retrieval, and Claude 4.5 Haiku inference via AWS Bedrock.

4.2. Results by Language and Task

4.2.1. Spanish Results

The Spanish test set, comprising 585 instances, was evaluated under both task configurations. Table 3 summarizes the prediction statistics.

Table 3

Prediction statistics for Spanish test set.

Task	Total	Reclaimed (1)	Not Reclaimed (0)	Avg. Confidence
Task A (Text Only)	585	196 (33.5%)	389 (66.5%)	0.861
Task B (Text + Bio)	585	200 (34.2%)	385 (65.8%)	0.860

For Task A, the system classified approximately one-third of the Spanish test instances as containing reclaimed language (33.5%), with the remaining two-thirds classified as not reclaimed. The average confidence score of 0.861 indicates relatively high certainty in the predictions, suggesting that the system could distinguish clear cases of reclamation from non-reclamation in the textual content.

The inclusion of biographical information in Task B resulted in a modest increase in reclaimed predictions, from 196 to 200 instances (a 2.0% increase in the proportion of positive predictions). This shift suggests that biographical context provided additional evidence supporting reclamatory interpretation for a small subset of instances that were ambiguous based on textual content alone. The average confidence remained stable at 0.860, indicating that the additional contextual information did not substantially alter the system’s overall certainty levels.

4.2.2. English Results

The English test set, containing 685 instances, was evaluated exclusively under Task A configuration, as biographical information was not available for this language. Table 4 presents the prediction statistics.

Table 4

Prediction statistics for English test set.

Task	Total	Reclaimed (1)	Not Reclaimed (0)	Avg. Confidence
Task A (Text Only)	685	67 (9.8%)	618 (90.2%)	0.892

The English test set exhibited markedly different prediction patterns compared to Spanish. Only 9.8% of instances were classified as containing reclaimed language, resulting in a highly imbalanced prediction distribution heavily skewed toward the negative class. This substantial difference from the Spanish predictions (33.5% positive) raises several interpretive considerations:

1. **Dataset characteristics:** The English test set may contain a genuinely lower proportion of reclaimed language instances, reflecting either sampling procedures or underlying differences in language reclamation prevalence across social media contexts in different languages.
2. **System conservatism:** The system may exhibit greater conservatism in classifying English instances as reclaimed, possibly due to the Knowledge Graph containing fewer English reclaimed terms, or due to differences in the retrieved training examples for English texts.
3. **Linguistic complexity:** English reclaimed language may present greater classification challenges due to more subtle contextual cues or greater ambiguity in term usage, leading the system to default to the negative class in uncertain cases.

Notably, the English predictions exhibited the highest average confidence score (0.892) across all languages and tasks, suggesting that despite the low positive prediction rate, the system expressed strong certainty in its classifications. This high confidence may indicate that many English test instances presented clear negative cases (e.g., neutral discussion of LGBTQ+ topics without reclaimed terminology), or alternatively, may reflect overconfidence in negative predictions.

4.2.3. Italian Results

The Italian test set, comprising 725 instances, was evaluated under both task configurations. Table 5 summarizes the prediction statistics.

Table 5

Prediction statistics for Italian test set.

Task	Total	Reclaimed (1)	Not Reclaimed (0)	Avg. Confidence
Task A (Text Only)	725	150 (20.7%)	575 (79.3%)	0.865
Task B (Text + Bio)	725	171 (23.6%)	554 (76.4%)	0.857

The Italian predictions fell between the Spanish and English extremes, with 20.7% of Task A instances classified as reclaimed. This intermediate proportion suggests either a genuinely moderate prevalence of reclamation in the Italian test set, or system behavior that lies between its conservative English performance and more liberal Spanish performance.

The addition of biographical context in Task B yielded a more substantial effect for Italian than for Spanish. The number of positive predictions increased from 150 to 171 (a 14.0% relative increase, or 2.9 percentage points in absolute terms). This suggests that Italian biographical information provided more disambiguating value than Spanish biographical information, possibly because Italian bios contained stronger community membership signals or because Italian textual content was more ambiguous in isolation.

The average confidence decreased slightly from 0.865 to 0.857 when biographical information was included, suggesting that while biographical context enabled additional positive classifications, it may have introduced some uncertainty or conflicting signals in certain cases.

4.3. Cross-Language Comparison

Figure 1 visualizes the prediction distributions across languages and tasks, facilitating direct comparison of system behavior. The visualization clearly demonstrates the substantial variation in classification patterns across the three target languages and the differential impact of biographical context.

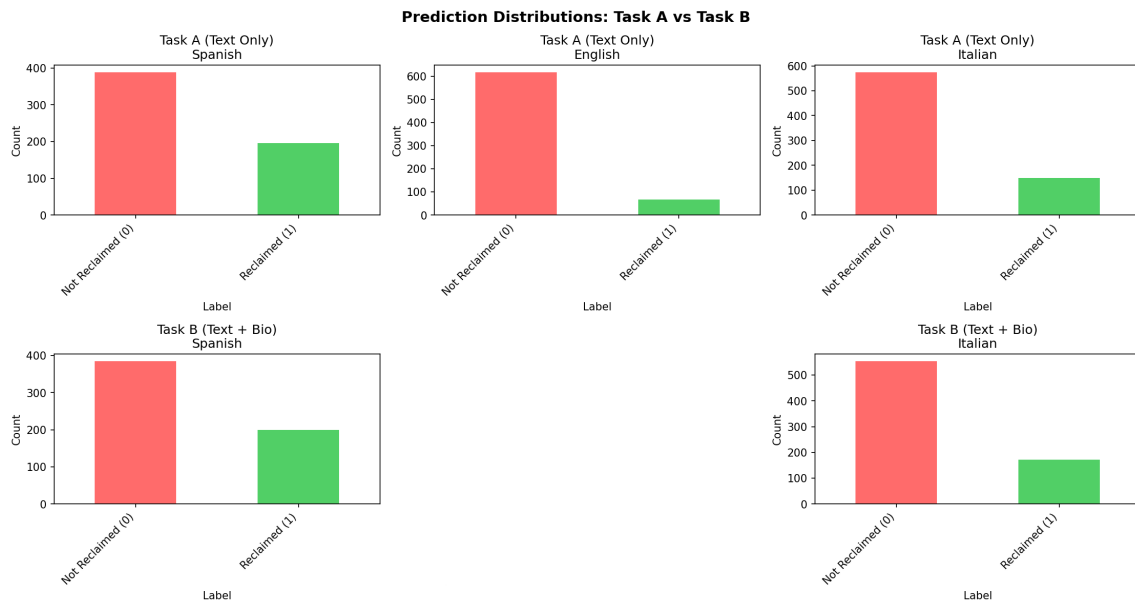


Figure 1: Prediction distributions for Task A and Task B across Spanish, English, and Italian test sets. Red bars represent not-reclaimed predictions (label 0), while green bars represent reclaimed predictions (label 1). The top row shows Task A results for all three languages, while the bottom row shows Task B results for Spanish and Italian (biographical information was not available for English).

4.3.1. Language-Specific Patterns

The visualization in Figure 1 reveals striking differences in prediction patterns across languages:

Spanish: Exhibits a relatively balanced distribution with approximately one-third positive predictions. The consistency between Task A and Task B suggests that textual content alone provides strong classification signals in Spanish, with biographical information playing a supplementary rather than transformative role. The visual similarity between the top-left (Task A) and bottom-left (Task B) panels confirms this minimal differential effect.

English: Shows an extremely imbalanced distribution with minimal positive predictions, as evidenced by the dominant red bar and minimal green bar in the top-middle panel. The stark contrast with Spanish and Italian suggests either genuine dataset differences or language-specific challenges in the classification system. The high confidence scores accompanying this imbalanced distribution warrant further investigation to distinguish between appropriate conservatism and potential systematic bias.

Italian: Demonstrates moderate prediction rates falling between Spanish and English, visible in the intermediate-sized green bar in the top-right panel. The visible increase in positive predictions from Task A (top-right) to Task B (bottom-right) indicates meaningful contribution of biographical context, suggesting that Italian reclaimed language classification benefits substantially from author-level information.

4.3.2. Task A versus Task B

The comparative analysis of Task A and Task B performance, readily apparent in Figure 1, provides insights into the value of biographical context:

Spanish: Minimal visual difference between the top-left and bottom-left panels (0.7 percentage points increase in positive predictions) suggests that Spanish textual content is largely self-sufficient for classification. This may reflect more explicit community markers or clearer contextual cues within Spanish tweets themselves.

Italian: The more noticeable difference between the top-right and bottom-right panels (2.9 percentage points increase) indicates meaningful contribution of biographical information. The visibly larger green bar in the Task B panel demonstrates that Italian reclaimed language usage may be more context-dependent, requiring author identity information to resolve ambiguity.

The modest reduction in confidence when biographical information is included (observed in both Spanish and Italian) may reflect increased complexity in the decision-making process. When biographical context conflicts with or complicates textual signals, the system appropriately expresses lower certainty.

4.4. Confidence Score Analysis

The average confidence scores across all tasks ranged from 0.857 to 0.892, indicating generally high certainty in classifications. Several patterns emerge:

- **English exhibits highest confidence:** The 0.892 average confidence for English, despite (or perhaps because of) the heavily imbalanced predictions visible in Figure 1, suggests either clear negative instances or potential overconfidence in negative classifications.
- **Biographical context reduces confidence:** Both Spanish and Italian show slight confidence reductions when biographical information is included (0.861 $\rightarrow$ 0.860 for Spanish; 0.865 $\rightarrow$ 0.857 for Italian), suggesting that additional context introduces complexity rather than clarity in some cases.
- **Moderate overall uncertainty:** While confidence scores are relatively high, they remain well below 1.0, indicating that the system recognizes the inherent difficulty of the task and expresses appropriate uncertainty rather than overconfident predictions.

5. Discussion

The experimental results reveal several important findings:

Cross-linguistic variation: The substantial differences in prediction rates across languages (9.8% for English, 20.7% for Italian, 33.5% for Spanish), clearly visible in Figure 1, highlight the language-specific nature of reclaimed language phenomena. These differences may reflect actual variation in reclamation prevalence, cultural differences in social media discourse, or differential system performance across languages.

Biographical context value: The comparison between Task A and Task B demonstrates that biographical information provides meaningful but not transformative value. The effect is more pronounced for Italian (14.0% relative increase in positive predictions) than Spanish (2.0% relative increase), as evidenced by the visual differences between panels in Figure 1.

System conservatism: Across all languages, the system tends toward conservative classification, with red bars consistently dominating green bars in the figure, potentially under-predicting reclamation rather than over-predicting it. This conservatism is most evident in English but present to varying degrees across all languages.

High confidence levels: The consistently high confidence scores (>0.85) suggest that the system identifies many clear-cut cases but may benefit from improved calibration to better distinguish varying levels of uncertainty.

These results provide a foundation for understanding system performance and identifying areas for future refinement, including enhanced cross-linguistic knowledge transfer, improved biographical context integration, and better calibration of confidence scores.

Acknowledgments

The authors wish to thank the support of AVAHL.

Declaration on Generative AI

During the preparation of this work, the authors used Writefull’s model in order to: Grammar and spelling check. Further, the authors used DeepL Translator in order to: Translate texts from Spanish to English.

References

- [1] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: <https://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, CoRR abs/1911.02116 (2019). URL: <http://arxiv.org/abs/1911.02116>. arXiv:1911.02116.
- [3] B. R. Chakravarthi, R. Priyadharshini, T. Durairaj, J. P. McCrae, P. Buitelaar, Overview of the shared task on homophobia and transphobia detection in social media comments, in: Proceedings of the Second Workshop on Language Technology for Equality, Diversity, and Inclusion (LT-EDI), 2022, pp. 1–6.

- [4] B. R. Chakravarthi, P. K. Kumaresan, R. Priyadharshini, P. Buitelaar, A. Hegde, H. Shashirekha, S. Rajiakodi, M. Á. García, S. M. Jiménez-Zafra, J. García-Díaz, R. Valencia-García, K. K. Ponnusamy, P. Shetty, D. García-Baena, Overview of third shared task on homophobia and transphobia detection in social media comments, in: *Proceedings of the Fourth Workshop on LT-EDI (EACL)*, 2024, pp. 1–10.
- [5] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: *PML4DC at ICLR 2020*, 2020.
- [6] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [7] T. Davidson, D. Warmesley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: *Proceedings of the 11th International AAAI Conference on Web and Social Media (ICWSM)*, 2017, pp. 512–515.
- [8] J. Kurrek, H. M. Saleem, D. Ruths, Towards a comprehensive taxonomy and large-scale annotated corpus for online slur usage, in: *Proceedings of the Fourth Workshop on Online Abuse and Harms, Association for Computational Linguistics*, 2020, pp. 138–149.
- [9] B. R. Chakravarthi, Detection of homophobia and transphobia in YouTube comments, *International Journal of Data Science and Analytics* 18 (2024) 49–68. doi:10.1007/s41060-023-00400-0.
- [10] B. R. Chakravarthi, R. Ponnusamy, S. Malliga, P. Buitelaar, M. Á. García-Cumbreras, S. M. Jiménez-Zafra, J. A. García-Díaz, R. Valencia-García, N. Jindal, Overview of second shared task on homophobia and transphobia detection in social media comments, in: *Proceedings of the Third Workshop on LT-EDI*, 2023, pp. 1–10.
- [11] J. Vásquez, S. Andersen, G. Bel-Enguix, H. Gómez-Adorno, S.-L. Ojeda-Trueba, HOMO-MEX: A Mexican Spanish annotated corpus for LGBT+phobia detection on Twitter, in: *Proceedings of the 7th Workshop on Online Abuse and Harms (WOAH)*, Association for Computational Linguistics, 2023, pp. 202–214.
- [12] G. Bel-Enguix, H. Gómez-Adorno, G. Sierra, J. Vásquez, S. T. Andersen, S.-L. Ojeda-Trueba, Overview of homo-mex at iberlef 2023: Hate speech detection in online messages directed towards the mexican spanish speaking lgbtq+ population, *Procesamiento del Lenguaje Natural* 71 (2023) 361–370.
- [13] H. Gómez-Adorno, G. Bel-Enguix, G. Sierra, J. Vásquez, S. T. Andersen, S.-L. Ojeda-Trueba, Overview of homo-mex at iberlef 2024: Hate speech detection towards the mexican spanish speaking lgbt+ population, *Procesamiento del Lenguaje Natural* 73 (2024) 393–405.
- [14] T. Davidson, D. Bhattacharya, I. Weber, Racial bias in hate speech and abusive language detection datasets, in: *Proceedings of the Third Workshop on Abusive Language Online*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 25–35. doi:10.18653/v1/W19-3504.
- [15] M. Sap, D. Card, S. Gabriel, Y. Choi, N. A. Smith, The risk of racial bias in hate speech detection, in: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 1668–1678.
- [16] S. T. Andersen, S.-L. Ojeda-Trueba, J. Vásquez, G. Bel-Enguix, The mexican gayze: A computational analysis of the attitudes towards the lgbt+ population in mexico on social media across a decade, in: *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH)*, Association for Computational Linguistics, 2024, pp. 178–200.
- [17] A. M. Davani, D. Kiela, M. Lambert, B. Xiang, B. Vidgen, T. Thrush, Z. Waseem, Harmful speech detection by language models exhibits gender-queer dialect bias, in: *Proceedings of the 4th ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, ACM, 2024, pp. 1–13.
- [18] T. D. Oliva, D. M. Antonialli, A. Gomes, Fighting hate speech, silencing drag queens? Artificial intelligence in content moderation and risks to LGBTQ voices online, *Sexuality & Culture* 25 (2021) 700–732. doi:10.1007/s12119-020-09790-w.

- [19] L. Dixon, J. Li, J. Sorensen, N. Thain, L. Vasserman, Measuring and mitigating unintended bias in text classification, in: Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, ACM, 2018, pp. 67–73.
- [20] S. L. Blodgett, S. Barocas, H. Daumé III, H. Wallach, Language (technology) is power: A critical survey of "bias" in NLP, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 5454–5476.
- [21] L. Draetta, S. Frenda, A. T. Cignarella, P. Viviana, ReCLAIM It! Exploring Italian slurs reappropriation with LLMs, in: Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2024), Pisa, Italy, 2024, pp. 1–7.
- [22] A. D. Galinsky, K. Hugenberg, C. Groom, G. V. Bodenhausen, The reappropriation of stigmatizing labels: Implications for social identity, in: J. T. Polzer (Ed.), Research on Managing Groups and Teams: Identity Issues in Groups, volume 5, Emerald Group Publishing Limited, 2003, pp. 221–256.
- [23] R. Brontsema, A queer revolution: Reconceptualizing the debate over linguistic reclamation, Colorado Research in Linguistics 17 (2004) 1–17.
- [24] L. Anderson, E. Lepore, Slurring words, *Noûs* 47 (2013) 25–48. doi:10.1111/j.1468-0068.2010.00820.x.
- [25] F. Elsafoury, Thesis distillation: Investigating the impact of bias in NLP models on hate speech detection, in: Proceedings of the Big Picture Workshop, Association for Computational Linguistics, Singapore, 2023, pp. 53–65. URL: <https://aclanthology.org/2023.bigpicture-1.5/>. doi:10.18653/v1/2023.bigpicture-1.5.
- [26] O. L. Haimson, D. Delmonaco, P. Nie, A. Wegner, Disproportionate removals and differing content moderation experiences for conservative, transgender, and black social media users: Marginalization and moderation gray areas, Proceedings of the ACM on Human-Computer Interaction 5 (2021) 1–35. doi:10.1145/3479610.

Challenger at MultiPRIDE: Is It Hate Speech or Reclaimed?

Hadi Bayrami Asl Tekanlou^{1,*}, Mahdi Bakhtiyarzadeh² and Jafar Razmara¹

¹Department of Computer Science, University of Tabriz, 29 Bahman Boulevard, Tabriz 51666-16471, Iran

²Department of Computer Science, University of Tabriz, 29 Bahman Boulevard, Tabriz 51666-16471, Iran

Abstract

The spread of hate speech has become increasingly harmful in modern digital environments, particularly on social networking platforms. While recent advances have shown promising results in automatic hate speech detection, a key challenge remains: distinguishing genuine hate speech from reclaimed language. Accurate labeling is difficult due to the nuanced and context-dependent nature of reclaimed expressions. In this paper, we present a simple and interpretable approach for distinguishing hate speech from reclaimed language, developed for the MultiPride Shared Task. Our method generates dense semantic text embeddings and incorporates a label-noise filtering stage using Cleanlab with logistic regression, followed by a Multi-layer Perceptron (MLP) neural network for final classification. The system is designed to operate under limited computational resources while maintaining strong performance. We evaluate our approach using precision, recall, and F1-score, including macro-averaged metrics. Experimental results demonstrate robust performance despite extreme class imbalance in the dataset. Overall, the findings highlight the potential for further improvements through larger embedding models and more advanced preprocessing techniques while preserving interpretability.

Content Warning: This paper contains examples of explicitly offensive content.

1. Introduction

The primary venue for communication between people is now the Internet. This has democratized online discourse, enabling users to share opinions on a vast array of topics. As a result, online platforms have become major spaces for public discourse, enabling rapid information exchange and large-scale interaction among users from diverse backgrounds. While this promotes free speech and broader participation, it also facilitates harmful behaviors when individuals target others based on religion, race, sex, sexual orientation, or political affiliation [1, 2, 3]. The use of such harmful or offensive expressions is referred to as hate speech. The exponential growth of social media platforms has facilitated this spread, transforming how hate speech disseminates through anonymity and rapid sharing [2, 4]. Hate speech, as a form of online abuse, has become a focal point in societal discussions, with computational tools emerging to mitigate its impact on individuals and communities [5]. On the opposite end of the spectrum, members of the same social group may use terms traditionally considered offensive as part of in-group appropriation or semantic reclamation [6]. This phenomenon, known as reclaimed language, presents a unique challenge in hate speech detection, where terms traditionally used derogatorily are repurposed by marginalized communities, such as the LGBTQ+ group, through mock impoliteness or semantic reclamation [6]. For instance, reclaimed slurs in in-group contexts may not convey hatred but instead foster solidarity; however, automated models often misclassify them as hateful, leading to false positives and restricting self-expression. It can be difficult to differentiate hate speech from other manifestations of potentially harmful language used within a particular group, as meaning depends strongly on speaker identity and contextual usage [7].

This study makes the following contributions:

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ h.bayrami1403@ms.tabrizu.ac.ir (H. B. A. Tekanlou); m.bakhtiyarzadeh1403@ms.tabrizu.ac.ir (M. Bakhtiyarzadeh); razmara@tabrizu.ac.ir (J. Razmara)

🌐 <https://github.com/HadiBayrami/> (H. B. A. Tekanlou); <https://github.com/Mahdi8424> (M. Bakhtiyarzadeh)

🆔 0009-0002-7206-6116 (H. B. A. Tekanlou); 0000-0002-6320-8517 (J. Razmara)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

- We begin by systematically cleaning the raw text data to reduce noise, filtering out URLs, non-ASCII characters (including emojis), and specific user mentions, while carefully stripping the # symbol from hashtags to retain their semantic textual content.
- To ensure reliable evaluation on our highly unbalanced dataset, we split the data into training and validation sets using stratified sampling, preserving the proportional distribution of labels across both subsets.
- We implement a comprehensive backtranslation augmentation strategy on the training data using the Google Translate API via `deep_translator`.
- We extract deep semantic features from the text samples by generating high-dimensional embeddings using the state-of-the-art `intfloat/e5-large-v2` model.
- We enhance the quality of our training signal by applying CleanLearning from the `cleanlab.classification` framework.
- Finally, we design and train a lightweight Neural Network specifically optimized for this task.

2. Related Work

As presented in the previous section, distinguishing between hate speech and reclaimed language is challenging due to the nature and linguistic complexity, since a clear boundary should be defined [6]. This problem stems from the contextual dependence of language, where marginalized groups, like the LGBTQ+ community, can use terms that are typically used as slurs to promote solidarity through "mock impoliteness" without expressing hatred. These subtleties are frequently overlooked by automated detection systems, which results in false positives that incorrectly label reclaimed language as hateful, restricting self-expression and possibly exacerbating discrimination [6, 7]. Biases in training datasets and classifiers, for example, make this problem worse because models trained on primarily out-group examples may overgeneralize slurs while neglecting in-group reclamation. Linguistic analyses also show how generalized forms of hate speech, which frequently rely on lethal or quantity-laden rhetoric, differ from directed forms, which are aggressive and personal. However, both can overlap with reclaimed uses in online contexts [8]. Early attempts to identify hate speech relied on rule-based keyword matching or simple machine learning classifiers. While these methods provided some level of detection, they struggled with context, ambiguity, sarcasm and had high error rates for sophisticated language [2]. Deep learning, with the introduction of semantic embeddings particularly using transformer models (e.g. BERT), has improved performance and made it easier to identify hateful intent across languages and domains by capturing bidirectional context. For example BERT was reweighted to reduce the racial bias in social media by accounting for African-American English (AAE) tweets being overclassified as offensive compared to Standard American English (SAE) tweets [7]. Despite these improvements, the models are still limited in their ability to handle reclaimed language due to their focus on speaker-group dynamics instead of surface level patterns [6]. Recent research shows just how crucial contextual information is for improving detection accuracy. When models take into account conversational threads or topic-specific metadata—for example, from Twitter—they can achieve up to 5.5 points higher Macro F1 scores in multi-label tasks, particularly for pandemic-related hate speech in Spanish varieties [9]. Context helps clarify meaning: a slur used within an in-group conversation may signal reclamation rather than aggression, and looking at replies or thread structure prevents interpreting posts in isolation. Psycholinguistic work on hate “lingo”—such as name-calling in targeted hate compared to more general religious or group-based rhetoric—also reinforces this need for context. Together, these insights support the development of hybrid models that blend linguistic features with training strategies that are aware of reclamation [8]. Strategies for bias mitigation are necessary for fair detection and to mitigate against inadvertent suppression of reclaimed language or expression. Techniques such as entropy-based regularization or sample reweighting in fine-tuning have shown efficacy at debiasing classifiers to attenuate the disproportionate flagging of AAE or LGBTQ+ discourse [7, 6]. The establishment of specialized datasets, including the Reclaimed Hate Speech Dataset (RHSD), will be essential to test models in an in-group setting and monitor for safeguards in semantic sensitivity

[6]. In addition to creating datasets, surveys of the landscape create a call for discourse within an interdisciplinary framework, to synthesize NLP with sociolinguistics, and enact easier boundaries between hate speech and reclamation, alongside the unraveling of online hate speech [2]. Still, research gaps remain: most datasets lack a rich compendium of distinct examples of reclamation, and a lack of cross-domain generalization to distinguish bias that can exist across varieties of English [7]. In the future, scholars working on the CMC dimension of hate speech should strive for datasets that are multilingual and intersectional, and that capture linguistic complexity in a global context, extending toward upholding justice to ensure models treat individuals belonging to marginalized groups without excessive over-moderation [6, 9, 2].

3. Task Description and Dataset

3.1. Task

MultiPRIDE shared task [10] aims to identify reclaimed language in LGBTQ+ situations on social media, specifically in tweets that use language tied to LGBTQ+ identity in a possibly harmful way or as language of reclamation. Thus, the task is to identify if it is used as reclaimed, so it is a binary classification task. The shared task is multilingual and includes datasets in Italian, Spanish, and English. The task provides two tracks:

- **Textual:** Participants are only granted access to the unprocessed text of each tweet. No additional metadata or contextually-relevant signals are provided.
- **Textual + Contextual:** Apart from the text, participants are free to rely upon optional contextual information related to the author, including profile description and any other relevant metadata.

Participants are welcome to examine not only models that rely strictly on the text of the tweet, but also models that consider additional contextual information if available (such as metadata regarding the author’s profile). This design facilitates the exploration of the effect contextual signals have on the interpretation of potentially sensitive terms.

3.2. Data

For this research, the data set was provided through the Multipride Shared Task: a collection of multilingual Twitter data, that has been annotated specifically for its use of reclaimed language. This data comprises English, Italian and Spanish, has been split into training and development, with each instance representing a tweet and being labelled either reclaimed or not reclaimed language – defining a binary classification scenario. Table 1 outlines how the instances are distributed across the three languages (English, Italian, and Spanish) in terms of their training or developmental status. Representative examples from the dataset are presented in Table 2 .

Table 1

Statistics of the experimental dataset.

Dataset	English	Italian	Spanish
Train	769	814	657
Development	257	272	219

4. Proposed Approach

For the MultiPride hate speech detection task, we constructed a pipeline that does not depend on any specific language (also referred to as language-agnostic). This pipeline includes several components:

Table 2

Representative examples from the dataset.

Lang	Biography	Tweet	Reclamation
En	N/A	I use the word tranny all the time...but that's only in reference to working on my cars. Transgendered, transvestite, and drag queen folk are too fabulous to have their descriptions abbreviated.	Yes
En	N/A	Actually that's what a faggot is. Fag is just something that needs to be burnt.	No
It	Certo le circostanze non sono favorevoli	In quanto disabile e frocia questi sono i miei PrideMonths. Ma vorrei anche dire che il giorno in cui nel manifesto di un evento lgbtqiatsfeminista verrà citato anche l'antiabilismo oltre ad anti sessismo/obtfobia/razzismo/specismo offro da bere	Yes
It	I veri partigiani furono i primi sovrani! w la patria!	Ecco, adesso pensate all'iter o in affitto ed al male che fate al bambino branco di finocchi arcobaleno	No
Es	Me llaman feminista, roja y bollera	Buenas tardes a rojos, feministas, republicanos, maricones, bolleras y demás LGTBI LGTBI pride	Yes
Es	I live for that energy!	Hace rato pasó una caravana de movimiento LGT...etc y algunos me vieron parado observando y me gritaron que yo también era marica. Órale, bien «respetuosas» estas personas que exigen respeto. PrideMonth Pride2022 LGTBQ	No

augmentation of training data, generation of robust embeddings through a neural network classifier, filtering of noisy labels and robust embeddings through a neural network classifier, and a neural network-based classification of hate speech detection, and all components of the pipeline are applied to all datasets (both English, Italian, and Spanish) consistently to mitigate the effect of class imbalance and the presence of noise in social media text.

4.1. Data Preprocessing and Augmentation

Data cleaning is a technique that utilizes raw text and removes sparsity and non-informative artifacts (non-informative means you cannot determine the origin of the content). For example, we remove URLs, user mentions (but keep the semantic meaning of hashtags), and non-ASCII characters (emojis, symbols, etc.) using regular expressions. We also apply a back-translation augmentation process to the hate speech datasets to address the significant class imbalance that exists. To do this, we take the hate speech minority class and translate it into multiple pivot languages (such as French, German, Russian, and Persian) and translate it back into the original source language using Google Translate API. This creates synthetic, positive samples of the minority class that enriches the training distribution and allows our models to generalize better among the minority class samples.

4.2. Feature Extraction and Label Cleaning

Text representations are created by the `intfloat/e5-large-v2` Sentence Transformer Model. The `intfloat/e5-large-v2` Sentence Transformer is a highly sophisticated model specifically designed to deliver semantic embeddings for input sentences. Input sequences are converted into long vector representations (a 1024-dimensional dense representation) that illustrate the minute bits of context found in the original messages that would not be picked up by a traditional lexical approach. Once the input sequences have been embedded, the initial training data is processed with Confident Learning (cleanlab) to ensure that the quality of the labels in the training data is valid. After the embedding process is completed,

Logistic Regression classifiers fit the embeddings into clusters based on the classifier identification of likely errors in the labels (label noise). Any instances classified as exhibiting a high incidence of label problems are removed from the training dataset in order to avoid introducing unwanted label noise during the subsequent final training phase.

4.3. Model Architecture and Optimization

Lightweight pipelines utilizing frozen pre-trained language model (PTLM) embeddings along with shallow neural sequence models have been investigated as part of previous studies related to hate speech detection [11]. In this article, authors analyze numerous minimalist architectures incorporated into a "shared task" (combining the use of orderless data) that's built on the combination of PTLMS and LSTMs, BiLSTMs, or transformers with a linear classifier. However, the framework in which we present our findings utilizes one cohesive pipeline created specifically for reclaiming language issues. The Multi-layer Perceptron (MLP) is made using the Tensorflow/Keras framework, and is referred to as the core classifier. The text embedding model we used is *intfloat/e5-large-v2* [12]. A multi-layer perceptron consists of an input layer that accepts the sentence embeddings, as well as two dense hidden layers (containing 16 and 8 units, respectively), both of which use the Rectified Linear Unit (ReLU) activation function. Finally, the output layer of the MLP uses the Sigmoid activation function to provide a binary probability score. The model is trained using the Adam optimizer and Binary Crossentropy loss function. Additionally, to compensate for class imbalance during training, class weights are computed and used to penalize misclassifications of the positive class more severely. The system architecture is illustrated in Figure 1

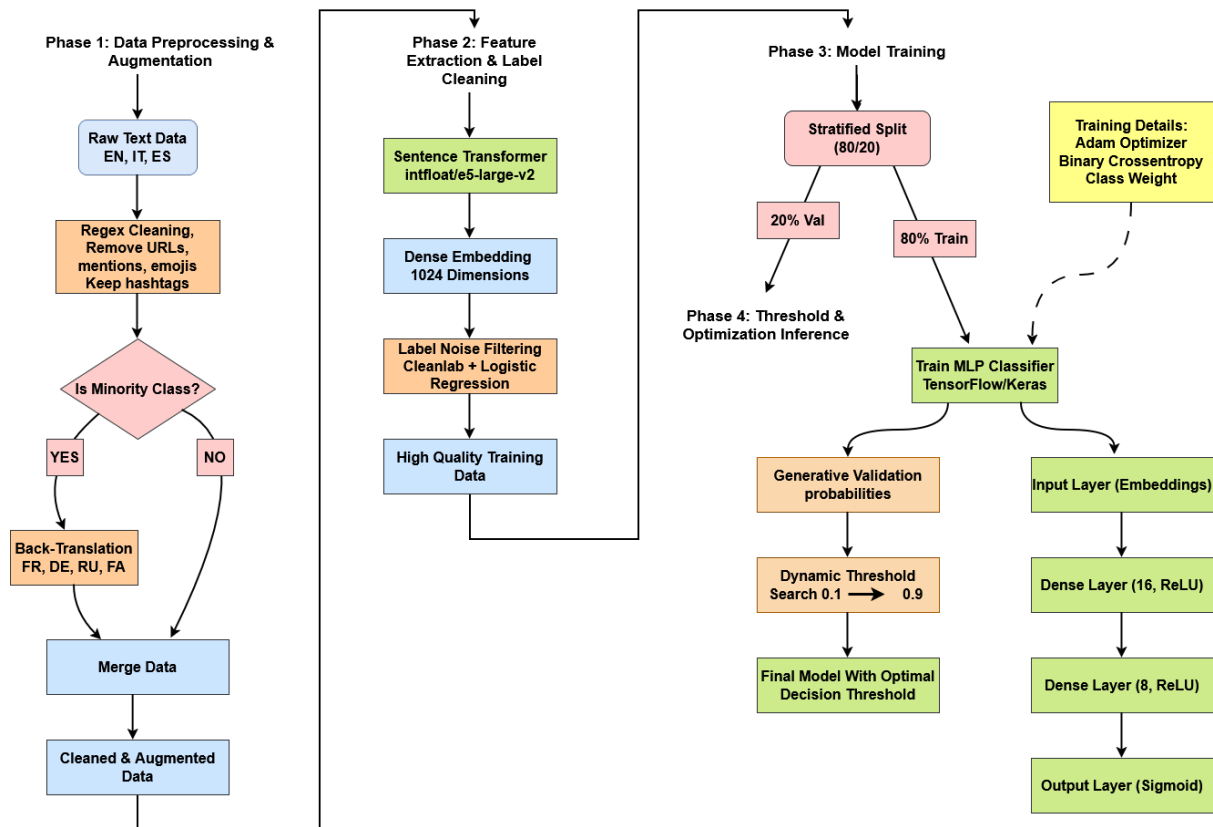


Figure 1: Overview of the proposed system architecture and training pipeline.

4.4. Validation and Threshold Tuning

In order to maintain representative class distributions throughout the development and evaluation phases, we utilized a 80/20 strata split when creating our training/validation sets. Additionally, after our model is trained, we used an adjusted threshold optimization method instead of simply applying a decision boundary of 0.5. In this method, our system has been designed to search through an entire range of decision boundaries (from 0.1 up to 0.9), and will select the threshold which produced the highest F1 score during validation in order to provide inferred outcomes from our test set using a threshold that was optimized for this project. Thus, we were able to optimize our model for the competition’s primary metric while maintaining the tradeoff between precision and recall.

5. Results

In this section, we describe the results obtained for multipride 2026. A total of 19 teams participated in the competition. The results presented in this paper are based on Precision, Recall, F1-score, and along with their macro-averaged variants. For this shared task baselines have not been reported. Consequently, the evaluation focuses exclusively on comparing the submitted systems based on the reported metrics, allowing for a direct assessment of each approach’s relative performance.

5.1. Evaluation Metrics

The criteria used to evaluate the submitted work reflect standard evaluation practices for assessing system performance and methodological soundness, and are as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

For any metric $M \in \{Precision, Recall, F1\}$, the macro-average is defined as:

$$Macro-avg(M) = \frac{M_{class\ 0} + M_{class\ 1}}{2}.$$

5.2. Team Results

The outcomes from our application of the proposed method we’ve described will now be discussed in detail. The performance of our system is assessed with the metrics shown in Section 5.1. As shown in the following Table 3, our system achieved the reported results. These results provide insight into the effectiveness of the proposed approach across different evaluation metrics.

Table 3
Classification performance per language

Language	Class 0			Class 1			Macro-avg		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
English	0.9355	0.9281	0.9318	0.2968	0.3220	0.3089	0.6162	0.6250	0.6203
Italian	0.9450	0.9692	0.9570	0.8548	0.7625	0.8060	0.8999	0.8659	0.8815
Spanish	0.9707	0.7344	0.8361	0.3684	0.875	0.5185	0.6695	0.8047	0.6773

As the main evaluation metric for system evaluation is macro F1-score, the corresponding bar plot highlighting the percentages is shown in Figure 2, which facilitates an easier comparison of results across languages and class labels.

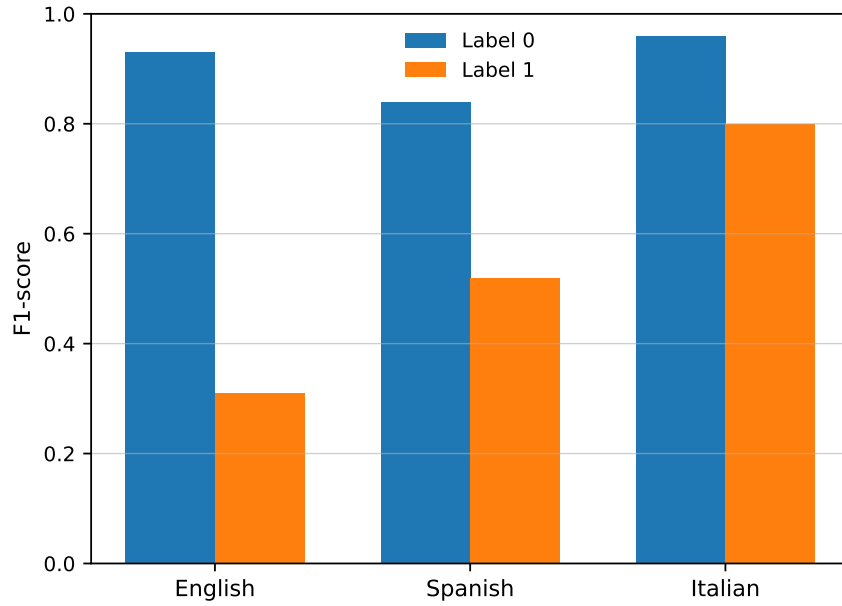


Figure 2: Comparison of F1-scores for binary classification labels (Label 0 and Label 1) across English, Spanish, and Italian.

6. Discussion

The findings presented herein indicate how efficiently the proposed technique operates in the context of datasets with significant amounts of data skewness towards one particular label or another based upon the assumption that some labels will appear more frequently than others. If implemented into operational environments today, such implementations would require minimal amounts of storage space for the trained model as well as only small amounts of computation time for each individual example within each batch of processed records. Thus, our results demonstrate that lightweight algorithms can effectively learn relationships between classes/labels based solely on a very small number of records within a given training set versus other possible approaches. In conclusion, the findings presented herein can be used as a starting point for additional research involving the development and implementation of lightweight models and algorithms that incorporate this approach. Although this methodology performed well on one labeled sample and had lower accuracy on the other labeled sample, this difference in performance is an indication of the significant class imbalance in the existing dataset. In fact, the results from this methodology were consistent with what is expected from an imbalanced learning environment, which does not indicate that the model was unstable. Also, the lower performance on a second labeled sample does not mean that the model is lacking capability but is indicative of the imbalanced data distribution and demonstrates that when looking at evaluation metrics, one must take into account the nature of the data.

7. Conclusions

Our team believes that utilizing a minimal alternative provides a new path for solving the problem of distinguishing between hate speech and reclaimed speech/language. This method provides an ideal balance between effectiveness and efficiency when dealing with limited training resources and computational capabilities. While our proposed approach performs well with limited resources, it has produced strong results across the evaluation criteria, although performance varies across languages; in particular, results for English were lower compared to other languages. Therefore, the applicability of this methodology across multiple languages demonstrates the potential strength and generalizability of our method in a multilingual environment. Moving forward, it will be valuable to utilize larger language models or generative models, as well as additional advanced methods and more complex

learning approaches, to address remaining classification difficulties and ultimately enhance the overall accuracy of the system.

Declaration on Generative AI

The application of generative artificial intelligence (AI) tools occurred only in a limited capacity to support the research undertaken in this study. For example, generative AI provided assistance with translating and interpreting meaning from the tweet dataset containing Italian language tweets as the authors of the study do not have fluency in Italian, being native speakers of English. In terms of facilitating improvement and clarity in the transcription of notes and written work generated by the authors, generative AI was also used in this process. Although generative AI was not applied towards generating, annotating, modelling or analysing any of the data collected or interpreting results from this study, all scientific/ statistical decisions and interpretations made throughout the study were conducted exclusively by the authors. After using these tools/services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] A. Albladi, M. Islam, A. Das, M. Bigonah, Z. Zhang, F. Jamshidi, M. Rahgouy, N. Raychawdhary, D. Marghitu, C. Seals, Hate speech detection using large language models: A comprehensive review, *IEEE Access* 13 (2025) 20871–20892. doi:10.1109/ACCESS.2025.3532397.
- [2] A. Schmidt, M. Wiegand, A survey on hate speech detection using natural language processing, in: L.-W. Ku, C.-T. Li (Eds.), *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, Association for Computational Linguistics, Valencia, Spain, 2017, pp. 1–10. URL: <https://aclanthology.org/W17-1101/>. doi:10.18653/v1/W17-1101.
- [3] A. Tontodimamma, E. Nissi, A. Sarra, L. Fontanella, Thirty years of research into hate speech: topics of interest and their evolution, *Scientometrics* 126 (2021) 157–179. URL: <https://doi.org/10.1007/s11192-020-03737-6>. doi:10.1007/s11192-020-03737-6.
- [4] A. B. De Oliveira, C. d. S. Baptista, A. A. Firmino, A. C. De Paiva, A large language model approach to detect hate speech in political discourse using multiple language corpora, in: *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing, SAC '24*, Association for Computing Machinery, New York, NY, USA, 2024, p. 1461–1468. URL: <https://doi.org/10.1145/3605098.3635964>. doi:10.1145/3605098.3635964.
- [5] A. Toktarova, D. Syrlybay, B. Myrzakhmetova, G. Anuarbekova, G. Rakhimbayeva, B. Zhylanbaeva, N. Suieuova, M. Kerimbekov, Hate speech detection in social networks using machine learning and deep learning methods, *International Journal of Advanced Computer Science and Applications* 14 (2023). URL: <http://dx.doi.org/10.14569/IJACSA.2023.0140542>. doi:10.14569/IJACSA.2023.0140542.
- [6] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: *Proceedings of the 16th ACM Web Science Conference (WEBSCI '24)*, ACM, Stuttgart, Germany, 2024, pp. 241–249. doi:10.1145/3614419.3644025.
- [7] M. Mozafari, R. Farahbakhsh, N. Crespi, Hate speech detection and racial bias mitigation in social media based on bert model, *PLOS ONE* 15 (2020) 1–26. URL: <https://doi.org/10.1371/journal.pone.0237861>. doi:10.1371/journal.pone.0237861.
- [8] M. ElSherief, V. Kulkarni, D. Nguyen, W. Y. Wang, E. Belding, Hate lingo: A target-based linguistic analysis of hate speech in social media, 2018. URL: <https://arxiv.org/abs/1804.04257>. arXiv:1804.04257.
- [9] J. M. Pérez, F. M. Luque, D. Zayat, M. Kondratzky, A. Moro, P. S. Serrati, J. Zajac, P. Miguel, N. Debandi, A. Gravano, V. Cotik, Assessing the impact of contextual information in hate speech detection, *IEEE Access* 11 (2023) 30575–30590. doi:10.1109/ACCESS.2023.3258973.

- [10] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [11] D. Nozza, A. Cignarella, G. Damo, T. Caselli, V. Patti, Hodi at evalita 2023: Overview of the first shared task on homotransphobia detection in italian, in: M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi (Eds.), Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), CEUR Workshop Proceedings, CEUR Workshop Proceedings (CEUR-WS.org), 2023. Publisher Copyright: © 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).; 8th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop, EVALITA 2023 ; Conference date: 07-09-2023 Through 08-09-2023.
- [12] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, F. Wei, Text embeddings by weakly-supervised contrastive pre-training, arXiv preprint arXiv:2212.03533 (2022).

DataSummit at MultiPRIDE: Context-Aware Multilingual Detection of Slur Reclamation in LGBTQ+ Contexts

Federico Dingeo, Marco Viviani*

Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo) – Edificio U14 (ABACUS), Viale Sarca, 336 – 20126 Milan, Italy

Abstract

The detection of hate speech in social media is a core challenge for Natural Language Processing (NLP) systems, particularly when offensive terms are used in non-harmful ways through the phenomenon of semantic reclamation. Reclaimed slurs, often employed by targeted communities as markers of identity and empowerment, pose significant difficulties for automatic moderation systems, which risk misclassifying legitimate content as abusive. This issue is especially relevant in multilingual and identity-rich contexts, such as discourse related to the LGBTQ+ community. In this work, the task of reclaimed slur recognition is addressed within the MultiPRIDE challenge at EVALITA 2026, focusing on Task B (Contextual Content), which incorporates user-level contextual information. A stratified classification approach that combines contextual embeddings from state-of-the-art multilingual Transformer models with engineered features derived from sentiment analysis, emotion detection, and user metadata is proposed. A multilingual training strategy is adopted to enhance cross-lingual generalization. Promising results were achieved during the development phase, showing the benefits of combining linguistic and metadata features. Nevertheless, the evaluation on the official test set revealed limitations in robustness. This highlights the difficulty of modeling semantic reclamation and the need for further improvements to ensure consistent performance on new data.

Keywords

Reappropriative intent, Semantic reclamation, LGBTQ+, Natural Language Processing, Text Classification

1. Introduction

In recent years, social media have emerged as one of the primary channels for written communication, making the detection of unmoderated content a central concern for *Natural Language Processing* (NLP) systems [1]. Within this landscape, the identification of *hate speech* and *abusive language* represents a particularly challenging task, as it requires a careful balance between limiting harmful content and safeguarding freedom of expression. A crucial aspect in this domain is the phenomenon of *semantic reclamation*, defined as the reappropriation of terms historically used as insults or slurs by the communities that were originally targeted by them [2]. In such cases, the offensive connotation of a term may be attenuated or transformed, acquiring instead a function related to identity affirmation and collective pride. NLP systems that fail to account for this phenomenon risk producing systematic misclassifications, ultimately penalizing the very groups they are designed to protect. Despite the extensive body of literature on *hate speech detection*, only a limited number of recent studies have explicitly addressed the distinction between “offensive use” and “reclaimed use” of slurs, such as [3]. This limitation becomes particularly salient in contexts characterized by a high density of identity-related language and slang, such as discourse within the LGBTQ+ community [4].

This work is conducted in the context of the **MultiPRIDE** challenge [5] at EVALITA 2026 [6], which focuses on the automatic recognition of reclaimed slurs in texts primarily extracted from Twitter. The task covers three languages—Italian, Spanish, and English—and concentrates on content related to the LGBTQ+ community. In addition to textual data, the dataset provided by the organizers includes contextual user information, when available, enabling a more nuanced modeling of legitimacy in linguistic

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ f.dingeo@campus.unimib.it (F. Dingeo); marco.viviani@unimib.it (M. Viviani)

🌐 <https://ikr3.disco.unimib.it/people/marco-viviani/> (M. Viviani)

🆔 0009-0000-3203-0894 (F. Dingeo); 0000-0002-2274-9050 (M. Viviani)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

usage. The objective of this study is twofold: to analyze the performance of different classification models on the reclaimed slur recognition task, and to contribute to a broader understanding of the challenges posed by semantic reclamation in multilingual NLP settings.

More specifically, our work addresses **Task B (Contextual Content)**, which leverages user biographical information to support classification decisions. To improve cross-lingual robustness, we adopted a multilingual training strategy aimed at enhancing model generalization. To tackle the task, we propose a stratified approach. As a *baseline*, we implement a Logistic Regression model based on TF-IDF representations. The core of the system, however, relies on contextual embeddings extracted from state-of-the-art multilingual Transformer models (*mDeBERTa V3*, *XLNet*, and *Multilingual MiniLM*), combined with linear classifiers. To better capture the semantic nuances associated with reclaimed language, we further enrich the vector representations through a *feature engineering* process that integrates signals derived from sentiment analysis, emotion detection, and metadata analysis (e.g., the use of emojis in user biographies). A subsequent feature selection phase is applied to reduce dimensionality and identify the most informative predictors. The robustness of the proposed system is assessed through a 10-fold cross-validation procedure on the training dataset.

The remainder of the paper is organized as follows: Section 2 introduces MultiPRIDE, describing its data and tasks; Section 3 presents the methodology adopted in this work, including exploratory data analysis, preprocessing pipeline, feature engineering strategies, and model architectures; Section 4 reports and discusses the experimental results; finally, Section 5 concludes the paper.

2. The MultiPRIDE Dataset and Tasks

This section presents the dataset released by MultiPRIDE and gives an overview of the proposed tasks, with a focus on the task considered in this work.

2.1. Description of the Dataset

The data collection provided for the MultiPRIDE task covers the period 2020–2022 and was constructed by integrating existing resources. The sources vary significantly depending on the language: while Italian and Spanish texts come exclusively from Twitter (sourced respectively from the TWITA collection and from specific LGBTQ+-themed datasets), the English subset was obtained by aggregating data from Twitter, Reddit, and TV series dialogues.

To build the final dataset, a common filtering process was applied across all languages. In the first phase, a keyword-based selection was performed using offensive terms (slurs) extracted from the HurtLex lexicon. Subsequently, the selection was refined to capture sentences with a high probability of reappropriative usage, filtering for positive terms expressing pride and community belonging (e.g., *pride*, *rainbow*).

The resulting dataset is multilingual and includes texts in Italian (1,086 samples), Spanish (876 samples), and English (1,026 samples). The three collections were merged into a single dataset containing 2,988 rows. Each instance in the dataset is associated with five variables:

- `id`: a unique identifier of the instance;
- `text`: the textual content;
- `bio`: the user biography, when available;
- `label`: the target label, a binary variable where 1 denotes the presence of words in reclaimed usage and 0 otherwise;
- `lang`: the language of the text.

2.2. Description of the Task(s)

The MultiPRIDE shared task comprises two binary classification tasks that aim to identify reappropriative intent in social media messages containing potentially derogatory terms related to the LGBTQ+ community.

- In **Task A**, systems are required to determine whether a message conveys reappropriative intent solely based on its textual content, i.e., considering only the message text;
- **Task B** extends this setting by permitting the use of contextual metadata associated with the author, specifically the biographical information provided in the *bio* field. This additional contextual information can assist in disambiguating between derogatory and reappropriative language use, enabling a more informed and accurate classification.

In this work, the focus is exclusively on **Task B**, as will be further clarified in the following Methodology section.

3. Methodology

This section describes the methodology adopted in this work. An exploratory analysis of the data is presented first, followed by the preprocessing steps applied to the dataset. Next, the feature engineering process and the model architectures used in the experiments are outlined.

3.1. Exploratory Data Analysis

Exploratory analysis proved essential to understanding the true nature of the data, identifying linguistic and contextual properties potentially relevant to the task, and to better prepare the next steps.

3.1.1. Assessing Data Quality and Class Distribution

A quick analysis of missing values revealed that the *bio* variable contains missing values for all English entries and for a portion of the Italian and Spanish entries (1,161 missing values in total). It is important to note that, although the task focused on exploiting biographical information, which is available only for Italian and Spanish, the English data were nevertheless included in the training phase. This strategy was followed with the aim of increasing the dataset size and helping the model generalize better by providing additional observations, even in the absence of biographical context.¹

Next, we checked for duplicate observations and removed five entries where the entire row (except for the *id*) was identical.

Inspecting the target variable (*label*), we observed a clear prevalence of Class 0 (non-reclaimed/offensive) over Class 1 (reclaimed). This imbalance is consistent across the three languages in the dataset and reflects the relative rarity of reclaimed uses compared to non-reclaimed ones. Such a distribution poses additional challenges for automatic recognition of the minority class, making the task sensitive to class imbalance.

3.1.2. Analysis of Texts and Biographies

For the token analysis, the texts and biographies were tokenized using NLTK's *TweetTokenizer*,² which is well-suited for social media content where hashtags, emojis, and slang are common.

Token counts per text show a positive skew, with most observations between 20 and 50 tokens. Stratifying by class, Class 0 texts tend to be longer and more dispersed, with numerous outliers, whereas Class 1 texts are shorter and more compact. This suggests that text length could be an informative feature for classification.

Biography lengths are more homogeneous, ranging from a few tokens up to about 40, and no relevant differences are observed between classes, indicating a lesser role for this feature.

We also examined the presence of biographies in relation to the target variable, excluding English texts due to complete missingness. Although minor percentage differences exist, no clear patterns suggest this feature as discriminative.

¹In light of the results actually obtained, as illustrated below, this solution may have proved suboptimal in retrospect due to potential meaning shifts or information loss, as also discussed in the limitations section.

²<https://www.nltk.org/api/nltk.tokenize.casual.html>

Additional stylistic features were considered. The proportion of fully uppercase words shows largely overlapping distributions between classes, with no significant differences in medians. Similarly, “aggressive punctuation”, i.e., defined as sequences of at least three consecutive ‘!’ or ‘?’, is rare and shows no meaningful class differences.

Overall, biography presence, uppercase usage, and aggressive punctuation do not provide sufficient evidence to warrant inclusion as primary features in the classification model.

3.1.3. Analysis of Hashtags and Emojis

Hashtags and emojis were analyzed using *Log-Odds* to identify discriminating terms for each class. Log-Odds measure how much a term is associated with one class versus the other by comparing its relative frequency in both groups. Positive values indicate a stronger connection to Class 1, while negative values indicate a stronger connection to Class 0.

It is important to note that the hashtag analysis was performed on the texts, while the emoji analysis was carried out on the biographies, with the goal of potentially identifying users belonging to the community more easily.

For Italian and English, no significant hashtags emerged. For Spanish, the term “orgullo” (pride) was strongly associated with Class 1. However, hashtags were **not** used as features in the final model, as the analysis showed high variability and inconsistent results: semantically similar hashtags sometimes had positive and sometimes negative Log-Odds values, even for those containing the term “orgullo”. This instability made the use of hashtags potentially unreliable and could have caused the model to overfit on specific keywords rather than learning the actual context of reclamation.

Regarding emojis in biographies, Class 1 users used emojis more frequently (~50%) compared to Class 0 users (~30%). In particular, three emojis showed a very strong association with the reclaimed class: the rainbow, the rainbow flag, and the transgender flag.

Table 1

Analysis of the top emojis in user biographies associated with the reclaimed class, based on Log-Odds.

Emoji	Prop_0	Prop_1	Log-Odds
	0.020	0.055	1.010
	0.088	0.187	0.759
	0.019	0.037	0.674

Note: Prop_0 and Prop_1 represent the relative frequency of the emoji within the respective class.

3.1.4. Analysis of Pronouns

Finally, the distribution of pronouns was analyzed using the *Stanza* library,³ focusing on pronouns associated with the subject or the verb to avoid double-counting. For each text, the number of occurrences of first-person expressions (singular or plural) and of second- or third-person expressions (singular or plural) was calculated. Based on these counts, one of the following labels was assigned to each text:

- **First person:** if the count of first-person occurrences is greater than that of other persons;
- **Neutral:** if the counts are equal;
- **Other persons:** if the count of second- and third-person occurrences is greater than that of the first person.

The analysis showed that texts belonging to Class 1 have a significantly higher proportion of cases with a predominance of first-person pronouns compared to Class 0. This finding supports the linguistic hypothesis of *reclamation*: when community members reclaim a slur, they tend to speak in the first person to express identity and belonging. In contrast, offensive or hate speech is more frequently expressed in the second or third person, reflecting distance or attack.

³<https://stanfordnlp.github.io/stanza/>

3.2. Text Preprocessing and Representation

In this stage, three distinct data cleaning operations were performed at different points in the workflow, each targeting a specific purpose and corresponding to different stages of text representation and preprocessing.

1. The **first preprocessing phase** was designed to make the *Exploratory Data Analysis* (EDA) more effective. In this phase, a minimal cleaning was applied: multiple spaces were reduced to a single space, and emojis were converted into a textual representation (*demojization*). This step was necessary because, during the EDA, it was observed that after tokenizing texts and biographies, the `TweetTokenizer` tended to split some emojis into sequences of multiple symbols, making analyses of token counts and emojis less interpretable;
2. The **second cleaning phase** was applied immediately before using pre-trained models to extract additional features, such as sentiment and emotion associated with the text. In this phase, HTML entities, user mentions, and URLs were handled; multiple spaces were also reduced to one;
3. The **third and final preprocessing phase** was applied before creating the text representations used for model training. This phase was divided into two parts, reflecting the two types of text representations used:
 - (a) The first representation is based on **TF-IDF** (*Term Frequency Inverse Document Frequency*), which, together with a Logistic Regression classifier, served as a *baseline*. Here, preprocessing was more extensive and aimed at noise reduction: all characters were converted to lowercase, URLs and mentions were removed, while for hashtags only the '#' symbol was removed, preserving the alphabetic part. All non-alphabetic characters were removed, and multiple spaces were reduced. After tokenization, stopwords were removed using the NLTK library, and the remaining tokens were lemmatized using SpaCy.⁴
 - (b) The second representation relies on **contextual embeddings**, obtained via three different models (*XLM-RoBERTa*, *mDeBERTa*, and *MiniLM*) and subsequently used with various linear classifiers. Preprocessing in this case was kept minimal, as excessive cleaning could have led to a loss of context and useful information; only the reduction of multiple spaces was applied.

3.3. Feature Engineering

This section describes the additional features derived from the available textual data, aimed at enriching the representation of instances beyond the text content alone. The considered features fall into two main groups: (i) features extracted using external pre-trained models, such as sentiment and emotions associated with the text, and (ii) surface features derived directly from the data, capturing structural, stylistic, and grammatical aspects.

3.3.1. Sentiment and Emotion Features

To extract these specific auxiliary features, the Italian and Spanish texts were translated into English, as the models used for sentiment and emotion analysis were trained exclusively on English data.⁵ In particular, the `GoogleTranslator` class from the `DeepTranslator` library was used for this step.⁶ All models employed for feature extraction were retrieved from the *Hugging Face* platform.

The feature related to **sentiment** was extracted using the *Twitter-RoBERTa Sentiment* model [7], based on the *RoBERTa-base* architecture. The model outputs a probability distribution over three classes (*i.e.*, *Negative*, *Neutral*, *Positive*) summing to 1. To ensure that only high-confidence predictions influenced the feature set, a thresholding logic was applied: if the confidence score of the predicted class was below

⁴<https://spacy.io/>

⁵It is important to note that this translation step was strictly limited to feature engineering; the core classification model relies on the original multilingual text processed via *XLM-RoBERTa* to preserve culture-specific nuances.

⁶<https://pypi.org/project/deep-translator/#google-translate-1>

0.5, the sentiment was remapped to *Neutral*, treating low-confidence predictions as non-informative. The results show a higher proportion of texts with positive or neutral sentiment in the minority class (Class 1), while Class 0 exhibits a clear prevalence of negative sentiment.

For **emotion** extraction, two complementary models were used. The first is the *DistilRoBERTa Emotion* model [8], which uses the *DistilRoBERTa* architecture. Trained on datasets including *TweetEval* and *Emotion*, it recognizes six basic emotions according to Ekman’s theory (i.e., *Anger*, *Disgust*, *Fear*, *Joy*, *Sadness*, *Surprise*) plus a *Neutral* category. This model, working on a limited number of less ambiguous categories, is generally more stable and precise. It shows that *Anger* and *Disgust* are significantly more frequent in Class 0, while *Joy* is more frequent in Class 1, consistent with the nature of the two classes.

The second model, *RoBERTa-Base GoEmotions*, is based on RoBERTa [9] and fine-tuned on Google’s *GoEmotions* dataset [10]. It handles colloquial language and direct opinions and classifies texts into 27 emotional labels, including *admiration*, *amusement*, *anger*, *annoyance*, *approval*, *caring*, *confusion*, *curiosity*, *desire*, *disappointment*, *disapproval*, *disgust*, *embarrassment*, *excitement*, *fear*, *gratitude*, *grief*, *joy*, *love*, *nervousness*, *optimism*, *pride*, *realization*, *relief*, *remorse*, *sadness*, and *surprise*, with a *neutral* category added. Results show higher proportions in Class 1 for *admiration*, *joy*, *love*, *pride*, and *sadness*, while *annoyance*, *anger*, *disapproval*, and *disgust* are more frequent in Class 0. This finer granularity captures detailed emotional aspects, showing that Class 1 is mainly associated with identity and belonging, whereas Class 0 contains more negative or judgment-related emotions.

3.3.2. Surface Features

Regarding the surface **additional features**, the *length of the text* and *length of the biography* were included, calculated as the number of tokens obtained via *TweetTokenizer*.⁷ Additionally, three binary variables were considered, indicating the presence or absence in the user’s biography of the following *emojis*: 🌈, 🇳🇱, and 🇧🇪. Finally, information about the *dominant pronoun* within the text was also included.

Although these are simple variables, their inclusion aims to provide the model with additional information that may help distinguish more easily between observations belonging to the two classes.

Conversely, variables related to the *presence of hashtags* containing the word “orgullo” and the *frequency of aggressive punctuation* within the text were not included. In the first case, as previously discussed (Section 3.1.3), this information was excluded to avoid biasing the model towards decisions based on single keywords, which could prevent it from capturing the overall context and generalizing effectively. In the second case, aggressive punctuation was not considered because the analyses showed that it was not a discriminating variable: except for very few cases, the majority of texts did not exhibit this characteristic.

3.4. Classification Models

This subsection details the experimental framework, starting with the establishment of a baseline model based on statistical text representations. The analysis then progresses to the evaluation of Transformer-based contextual embeddings, followed by the specific procedures adopted for feature engineering, dimensionality reduction, and final hyperparameter tuning. Crucially, across all configurations, every model was trained with the parameter `class_weight=‘balanced’` to address the significant class imbalance present in the data.

3.4.1. Baseline Model

As a baseline, it was decided to train a Logistic Regression classifier [11] on a TF-IDF (Term Frequency–Inverse Document Frequency) representation [12] of the textual content. To incorporate contextual information as required for Task B, the user biography was concatenated to the text (separated by a space) to create a single input string for feature calculation.

⁷<https://www.nltk.org/api/nltk.tokenize.casual.html>

Since this step serves as a simple starting point, standard parameters were used: unigrams, removal of rare terms (present in fewer than 5 documents) and highly frequent terms (present in more than 90% of documents), and a maximum limit of 5,000 features. This ensures a balance between the information used and model simplicity.

As described in Section 3.2, a noise-reduction preprocessing phase was required to obtain this representation: converting text to lowercase, removing URLs and mentions, deleting the ‘#’ symbol from hashtags, reducing multiple spaces, removing stopwords, and performing lemmatization.

As a final step, the TF-IDF vectors were used as features to train a Logistic Regression classifier, with *class_weight* setted to *balanced* in order to mitigate the class imbalance issue, which was then applied to classify each observation into one of the two classes. The classifier was evaluated using 10-fold cross-validation to better estimate its generalization performance.

This approach provides a simple, fast, and interpretable baseline, which is essential for a clear comparison with subsequent models based on contextual embeddings.

3.4.2. Contextual Embeddings

In this phase, different contextual embeddings, obtained via three pre-trained models, were tested on three linear classifiers: Logistic Regression, Linear SVM [13], and Ridge Classifier [14]. The models used to extract the embeddings were *mDeBERTa*, *XLM-RoBERTa*, and *MiniLM*, all belonging to the Transformer family.

The choice of these models was driven by the desire to compare solutions with different characteristics, all capable of producing contextual embeddings. *XLM-RoBERTa* and *mDeBERTa* are “strong” multilingual models, often used as benchmarks in multi-language scenarios, and are thus suitable for handling a dataset that includes Italian, Spanish, and English without having to train a separate model for each language. Conversely, *MiniLM* was selected as a more compact and faster alternative to evaluate whether a lighter model could still capture enough context to distinguish between reclaimed and non-reclaimed use of terms.

The goal of this phase was to identify which of the three textual representations performed best on the available data, in order to focus on a single one for subsequent improvement steps. For this reason, tests were conducted using exclusively embeddings extracted from the text (`text` column), excluding biography information to evaluate the potential of the three representations more clearly. Furthermore, the representations were obtained by applying *mean pooling* to the contextual token vectors; this choice allowed for a simple and stable single text representation without capturing overly specific features, making the comparison between models more direct.

In this comparison, the embedding extracted with *XLM-RoBERTa* achieved systematically better results than the other two, regardless of the linear classifier used, based on a 10-fold cross-validation evaluation.

3.4.3. XLM-RoBERTa Representation

After observing that the representation extracted via *XLM-RoBERTa* achieved better results with all tested classifiers, even when limited to text alone, the data representation was enriched. First, the embedding representation was modified by concatenating the *mean pooling* vector with the *max pooling* vector. The rationale is that these two methods capture different and partially complementary information: *mean pooling* summarizes the average text content and provides a more stable representation, while *max pooling* highlights dimensions where strong signals appear, often linked to particularly informative tokens.

Subsequently, the user biography representation was added, using the same procedure as for the text (*XLM-RoBERTa* plus concatenation of *mean pooling* and *max pooling*). This step is central to Task B, as it allows for the inclusion of the author’s context, thereby integrating the information provided in the text more effectively.

Finally, to further increase the signals available to the classifier, features obtained from the feature engineering phase were concatenated. Features extracted through pre-trained models regarding sentiment (*Twitter-RoBERTa Sentiment*) and emotion (*DistilRoBERTa Emotion* and *RoBERTa-Base GoEmotions*) were introduced to add high-level information that the model can exploit directly. Manually constructed features included token counts for both text and biography, variables indicating the presence in the biography of emojis strongly associated with the reclaimed class (🏳️, 🏳️‍🌈, and 🏳️‍🚫), and information about the dominant pronoun in the text. These features were intended to add signals related to the structure, grammar, and style of the data.

A transformation was applied to all these variables to make them directly usable by the model. Specifically, continuous variables were normalized within the $[0, 1]$ interval, categorical variables were coded using *one-hot encoding*, and binary variables were left unchanged.

3.4.4. Feature Selection and Hyperparameter Tuning

After defining the representations for text, biography, and additional features, a *feature selection* phase was performed to identify the most informative variables and reduce dimensionality. This step was essential to decrease computational training costs and eliminate redundant or uninformative features, with the aim of improving the model's predictive performance for the binary classification task.

Specifically, the initial feature set was reduced from 3,115 to 800. The 3,115 starting features were composed as follows:

- 1,536 features related to the text and 1,536 related to the user biography. Following the requirements for Task B, which encourages leveraging contextual information, both representations consist of the concatenation of 768 dimensions extracted from *XLM-RoBERTa* via *mean pooling* and 768 via *max pooling*.
- 43 additional features obtained during the feature engineering phase.

The reduction was carried out using *Recursive Feature Elimination* (RFE) [15], utilizing the implementation available in the `scikit-learn` library.⁸ This method iteratively removes the least relevant features based on the model's coefficients. As a result, the usefulness of the engineered features—such as emoji indicators—was implicitly evaluated, keeping only those that added significant value to the contextual embeddings.

Since the linear *Support Vector Machine* (SVM) was observed to be the best-performing classifier during this phase, the RFE parameters were also optimized—specifically `n_features_to_select`, representing the final number of features, and `step`, indicating the number of features to remove at each iteration.

Finally, a search for the best hyperparameters for the SVM classifier was conducted, considering both the *linear* and *Radial Basis Function* (RBF) kernels, along with the regularization parameter C . Specifically, values for C of 0.1, 1, 10, and 100 were tested. The experimental results indicated that the best-performing configuration coincided with the default settings: a *linear* kernel and $C = 1$.

3.4.5. Final Training on the Dataset

Up to this point, the various configurations of data representations and classifiers have been evaluated using 10-fold cross-validation on the provided training set. Since a labeled test set was not available for local evaluation during the development phase, as the official test set is released without labels for the challenge assessment, cross-validation was preferred over a standard training/test split. This approach provided a more stable performance estimate and reduced the risk of results being overly dependent on a single data partition.

Consequently, in this final phase, textual representations for all observations were constructed using *XLM-RoBERTa*, concatenating the vectors obtained via *mean pooling* and *max pooling*. The features identified during the feature engineering stage were added to these representations, specifically

⁸https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

incorporating contextual information like biographies as required for Task B. Subsequently, only the features identified during the previous *feature selection* phase were retained to maintain the optimized dimensionality of 800 features.

Finally, the linear SVM classifier was trained on all available observations in the dataset. This final model was used to generate the predictions submitted for the official evaluation of the **MultiPRIDE** challenge.

4. Results

The following section presents the experimental results obtained during the development and evaluation phases. The analysis is divided into two main parts. First, the performance on the labeled training dataset is analyzed to compare different model configurations and select the best approach. Second, the results obtained on the official blind test set are reported, evaluating the final model’s effectiveness on unseen data.

4.1. Labeled Dataset

The experiments described in this section were conducted using the labeled training dataset provided by the MultiPRIDE challenge organizers. To rigorously assess model performance and generalization capability during the development phase, a 10-fold cross-validation strategy was applied across all reported configurations.

Table 2 compares the TF-IDF baseline combined with Logistic Regression against various setups based on contextual embeddings from pre-trained models, paired with different classifiers, considering exclusively the text. In this phase, designed to identify the most suitable model for the case study and lacking additional contextual information, the embedding-based solutions do not show an improvement over the baseline.

Table 2

10-fold Cross-validation results for different model configurations. Bold values indicate the best performance.

Model Configuration	F1-score Class 0	F1-score Class 1	Macro F1-score
Baseline: TF-IDF + Logistic Regression	0.9123	0.5690	0.7406
mDeBERTa + Logistic Regression	0.8872	0.5074	0.6973
MiniLM + Logistic Regression	0.8655	0.5186	0.6920
XLM-RoBERTa + Logistic Regression	0.8791	0.5186	0.6988
mDeBERTa + Linear SVM	0.8782	0.4670	0.6726
MiniLM + Linear SVM	0.8853	0.5235	0.7044
XLM-RoBERTa + Linear SVM	0.8551	0.4612	0.6581
mDeBERTa + Ridge Classifier	0.8832	0.4834	0.6833
MiniLM + Ridge Classifier	0.8627	0.4756	0.6691
XLM-RoBERTa + Ridge Classifier	0.8915	0.5371	0.7143

Note: The embeddings used are generated solely via mean pooling and extracted only from the text data.

Table 3 reports the results obtained using the representation based on XLM-RoBERTa (on text and biography), to which the engineered features were concatenated, in combination with various classifiers. Considering the modest performance observed in the previous phase, an improvement over the baseline was not expected at this stage: the overall representation comprises 3,115 features, a high number that can introduce significant noise and redundancy, increase training times, and penalize generalization capabilities, resulting in suboptimal performance.

Table 4 presents the results obtained by applying feature selection via RFE to the same representation described in Table 3. Even with RFE(300), an improvement over the baseline is observed, indicating that

Table 3

10-fold Cross-validation results for different model configurations. Bold values indicate the best performance.

Model Configuration	F1-score Class 0	F1-score Class 1	Macro F1-score
Baseline: TF-IDF + Logistic Regression	0.9123	0.5690	0.7406
XLM-RoBERTa + Logistic Regression	0.9067	0.5257	0.7162
XLM-RoBERTa + Linear SVM	0.8949	0.4236	0.6592
XLM-RoBERTa + Ridge Classifier	0.8902	0.4418	0.6660

Note: The representation is obtained by concatenating the mean- and max-pooled embeddings from the text and biography fields (processed identically), and then appending the engineered features extracted during the feature engineering step.

the previous representation was not uninformative but rather penalized by a high level of noise and redundant features. The three configurations using RFE(300) were used to identify the most suitable classifier for this representation, which proved to be the Linear SVM. Once the classifier was fixed, further experiments showed that the overall best configuration is obtained by selecting 800 features (RFE(800)).

Table 4

10-fold Cross-validation results for different model configurations. Bold values indicate the best performance.

Model Configuration	F1-score Class 0	F1-score Class 1	Macro F1-score
Baseline: TF-IDF + Logistic Regression	0.9123	0.5690	0.7406
XLM-RoBERTa + RFE(300) + Logistic Regression	0.9250	0.6522	0.7886
XLM-RoBERTa + RFE(300) + Linear SVM	0.9428	0.7384	0.8406
XLM-RoBERTa + RFE(300) + Ridge Classifier	0.9371	0.7052	0.8211
XLM-RoBERTa + RFE(800) + Linear SVM	0.9608	0.8019	0.8813

Note: The base representation is the same as in Table 3. Feature selection is performed using RFE (Recursive Feature Elimination); RFE(k) denotes selecting k features.

4.2. Test Set

The results presented in this section were obtained using the optimal configuration identified during the development phase. The final model relies on contextual representations extracted via XLM-RoBERTa from both the text and user biography, aggregating mean-pooled and max-pooled vectors. To optimize performance, Recursive Feature Elimination (RFE) was applied to select the top 800 features. A Linear SVM classifier (regularization parameter $C = 1$) was then trained on the entire available labeled dataset to generate predictions for the blind test set.

Since the test set labels were not available, the evaluation was conducted exclusively based on the metrics provided by the challenge organizers following the submission of predictions.

Table 5

Classification report results for the Italian dataset.

Category	Precision	Recall	F1-Score
Class 0	0.9117	0.9163	0.9140
Class 1	0.6397	0.6258	0.6327
Macro Avg	0.7757	0.7711	0.7733

Table 6

Classification report results for the Spanish dataset.

Category	Precision	Recall	F1-Score
Class 0	0.9050	0.8249	0.8631
Class 1	0.3409	0.5113	0.4090
Macro Avg	0.6229	0.6681	0.6361

The results show a clear difference between the two languages: for observations with Italian text, performance appears generally more stable, whereas for Spanish text, a substantial drop is observed, mainly attributable to greater difficulty in recognizing the minority class.

In both cases, however, performance on the test set is significantly lower than that observed on the available labeled data. This gap suggests that the model overfitted the training set, consequently reducing its ability to generalize to unseen data.

5. Conclusion

In the context of the MultiPRIDE challenge (EVALITA 2026), the test set was unlabeled; consequently, the final evaluation was derived exclusively from the metrics provided by the organizers based on the submitted predictions. Given the class imbalance, Macro F1 was used as the primary metric. The objective of this section is to summarize the experimental process and justify the choice of the final combination of data representation and model.

In the initial phase, a baseline based on TF-IDF and Logistic Regression was defined and subsequently compared with approaches leveraging contextual embeddings extracted from pre-trained models combined with various classifiers (Logistic Regression, Linear SVM, and Ridge Classifier). Considering exclusively the text, and thus without using contextual information (user biography), these setups did not show improvements over the baseline; however, they were useful primarily for guiding the choice of data representation for the current task.

Subsequently, a more comprehensive representation was introduced, obtained by concatenating text and biography representations (mean and max pooling) and integrating features derived from feature engineering. In this configuration, however, the high overall dimensionality amplified noise and redundancy, negatively affecting both training times and performance, which did not surpass the baseline.

Improvement was observed with the introduction of feature selection via Recursive Feature Elimination (RFE): even with an initial selection of features, results superior to the baseline were achieved, indicating that the extended representation was informative but penalized by the presence of many non-useful features. Comparisons between classifiers in the RFE configuration highlighted the Linear SVM as the most effective choice; once the classifier was fixed, further experiments showed that the absolute best configuration is obtained by selecting 800 features.

Finally, a drop in performance is observed on the test set compared to estimates obtained via cross-validation on the available labeled data, signaling weak generalization and thus overfitting. Furthermore, a difference between languages emerges: results are generally more stable on Italian texts, while on Spanish texts the decline is more marked, especially for the minority class. Since test set labels are not available, an in-depth error analysis cannot be performed; however, the gap between validation and test results may be linked to differences between training and test data, to an overly aggressive optimization of Macro F1 on the available data, which may reduce robustness on unseen data, or to the fact that the multilingual model was unable to achieve an equally high level of generalization across different languages, specializing better in one than in the others.

5.1. Identified Limitations

A first limitation concerns the quantity of available data. After removing duplicates, the training set contained 2,983 total samples distributed across the three languages (Italian, Spanish, and English). Consequently, the number of examples per single language was relatively low, with a potential impact on the model’s ability to generalize.

A second limitation is related to finding the configuration that maximized the Macro F1 score on the available data. Although useful for quickly comparing many solutions, this approach may have favored choices that were too “fitted” to the training set, increasing the risk of overfitting and reducing robustness on unseen data.

Furthermore, the evaluation was conducted by considering all languages together, without systematically analyzing performance language-by-language during model selection. This made it harder to detect potential imbalances between languages in advance, which instead emerged clearly during the testing phase (for example, the more marked drop in performance for Spanish).

Finally, the presence of English in the training phase may have introduced meaning shifts or information loss: the model was trained in a multilingual setting on three languages, but the final evaluation concerned only two languages. Additionally, the English texts did not include the biography. This misalignment (same model, but different information depending on the language) could have added noise and made learning less effective, especially for linear classifiers.

5.2. Future Developments

First of all, it would be useful to directly address the identified limitations: increasing evaluation robustness (e.g., by analyzing performance per language during cross-validation), reducing the risk of overfitting (using more conservative selection and tuning strategies), and managing differences between languages and biography availability more consistently.

A natural second step is to test other multilingual models for extracting contextual embeddings, verifying whether different representations prove to be more stable across languages and more informative for the minority class.

Finally, an interesting approach to explore is zero-shot classification. In this scenario, the model is not directly trained on the task with labeled examples but attempts to classify texts using a class description (or a prompt) and the knowledge already acquired during pre-training. This can be useful when labeled data is scarce or when the goal is to improve cross-lingual generalization without relying too heavily on the specific training set.

Code and Data Availability

For reproducibility purposes, the source code is publicly available at: <https://github.com/FedericoDingeo/Multipride-at-EVALITA-2026-Project/>. To obtain access to the data, we invite the reader to refer to the guidelines published in [5].

Declaration on Generative AI

During the preparation of this work, the authors employed ChatGPT, Gemini, and Grammarly, to assist with text revision, primarily for grammar and spell checking, paraphrasing, and enhancing the clarity and style of the English writing. Specifically, these tools were utilized to refine the phrasing of certain passages and concepts, and to assist in translating ideas into clearer English formulations. All outputs produced by these tools were carefully reviewed and, where necessary, edited by the authors, who take full responsibility for the final content of the paper.

References

- [1] Y. Goldberg, *Neural Network Methods for Natural Language Processing*, Morgan & Claypool Publishers, 2017.
- [2] R. Brontsema, A queer revolution: Reconceptualizing the debate over linguistic reclamation, *Colorado Research in Linguistics* 17 (2004) 1–17. URL: <https://journals.colorado.edu/index.php/cril/article/view/238>.
- [3] L. Draetta, C. Ferrando, M. Cuccarini, L. James, V. Patti, ReCLAIM project: Exploring Italian slurs reappropriation with large language models, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 335–342. URL: <https://aclanthology.org/2024.clicit-1.40/>.
- [4] L. Dixon, J. Li, J. Sorensen, N. Thain, L. Vasserman, Measuring and mitigating unintended bias in text classification, in: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 2018, pp. 67–73. URL: <https://doi.org/10.1145/3278721.3278729>.
- [5] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [6] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [7] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke, L. Neves, TweetEval: Unified benchmark and comparative evaluation for tweet classification, in: *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, 2020, pp. 1644–1650. URL: <https://aclanthology.org/2020.findings-emnlp.148>. doi:10.18653/v1/2020.findings-emnlp.148.
- [8] J. Hartmann, Emotion english distilroberta-base, 2022. URL: <https://huggingface.co/j-hartmann/emotion-english-distilroberta-base/>.
- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, *arXiv preprint arXiv:1907.11692* (2019). URL: <https://arxiv.org/abs/1907.11692>.
- [10] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. MonTERSO, S. Ravi, GoEmotions: A dataset of fine-grained emotions, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Online, 2020, pp. 4040–4054. URL: <https://aclanthology.org/2020.acl-main.372/>.
- [11] D. R. Cox, The regression analysis of binary sequences, *Journal of the Royal Statistical Society: Series B (Methodological)* 20 (1958) 215–232.
- [12] K. Spärck Jones, A statistical interpretation of term specificity and its application in retrieval, *Journal of documentation* 28 (1972) 11–21.
- [13] C. Cortes, V. Vapnik, Support-vector networks, *Machine learning* 20 (1995) 273–297.
- [14] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, *Technometrics* 12 (1970) 55–67.
- [15] M. Kuhn, K. Johnson, *Applied Predictive Modeling*, Springer, New York, 2013.

Ghavidel-Rajabi at MultiPRIDE: Identity, Toxicity, or Complexity? A Language-Specific Feature Selection Approach to Reclamatory Intent Detection

Houman Rajabi^{1,*†}, Fatemeh Ghavidel^{1,*†} and Kouros Ghahremani^{2†}

¹Department of Computer Science, University of Turin, Corso Svizzera 185, 10149 Turin, Italy

²Faculty of Humanities, Islamic Azad University, Hamedan Branch, Professor Mousivand Blvd, Madani Town, 65181-15743 Hamedan, Iran

Abstract

Detecting reclamatory intent—the empowering use of potentially offensive terms within marginalized communities—remains a complex challenge in multilingual hate speech detection. This paper presents a Hybrid Fusion architecture developed for the MultiPRIDE @ EVALITA2026 shared task (Task A), predicated on the hypothesis that the linguistic markers of reclamation vary significantly across cultures. Unlike generic multilingual baselines, our approach decouples feature selection for English, Spanish, and Italian. By fusing monolingual BERT embeddings with language-optimized feature vectors, we isolate distinct "drivers" of reclamation: syntactic complexity in Italian, sentiment-toxicity gaps in Spanish, and identity-based pronoun usage in English.

The proposed pipeline utilizes Random Forest to prioritize the most predictive sociolinguistic features, yielding Macro F1-scores of 0.8981 for Italian and 0.6960 for Spanish, but only 0.5460 for English. Our results demonstrate that while explicitly modeling sociolinguistic nuance drives highly competitive performance in Romance languages, the English results reveal a critical "Identity Trap." Our error analysis demonstrates that the model's reliance on first-person pronouns fails to distinguish empowering reclamation from the reporting of victimization. Lacking the semantic depth to resolve this ambiguity, the English model collapses into a hyper-conservative classifier with negligible recall, highlighting that surface-level features are insufficient for English and necessitating semantic role labeling to distinguish agents from targets.

Keywords

MultiPRIDE, Reclamatory Intent Detection, Language-Specific Feature Selection, Hybrid Fusion, LGBTQ+ Discourse

1. Introduction

The automated detection of hate speech has become a cornerstone of content moderation, yet it frequently struggles with the nuances of reclaimed language—the phenomenon where marginalized communities reappropriate pejorative terms for self-empowerment. In LGBTQ+ discourse, terms like "queer" or "faggot" function simultaneously as slurs when used by outsiders and as badges of identity when used by insiders. This duality creates a "reclamation paradox" where the semantic toxicity of a word contradicts the pragmatic intent of the speaker. For standard Natural Language Processing (NLP) models, which often rely on lexical cues and surface-level toxicity scores, these "bad words in good contexts" create a high risk of false positives, silencing the very voices they are meant to protect [1, 2].

While recent advancements in multilingual Transformers (such as XLM-RoBERTa) have improved generalized sentiment analysis, scholarship suggests they often fail to capture culture-specific pragmatics [3]. A "one-size-fits-all" multilingual approach implicitly assumes that the linguistic markers of empowerment are universal. However, research in cross-lingual transfer indicates that models struggle

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, February 26–27, 2026, Bari, Italy

*Corresponding author.

†These authors contributed equally.

✉ houman.rajabi@edu.unito.it (H. Rajabi); fatemeh.ghavidel@edu.unito.it (F. Ghavidel); kouros.ghahremani@yahoo.com (K. Ghahremani)

🌐 <https://houmanrajabi.github.io/> (H. Rajabi)

🆔 0009-0005-2484-2442 (H. Rajabi); 0009-0003-9427-6592 (F. Ghavidel); 0009-0006-0890-7231 (K. Ghahremani)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

with tasks requiring deep sociolinguistic competence, often defaulting to a “Western” or Anglicized view of discourse [4, 5].

In this paper, we present a Hybrid Fusion Architecture developed for the MultiPRIDE @ EVALITA2026 shared task (Task A) [6], designed to resolve this paradox. To overcome the “curse of multilinguality”—where massive models dilute performance on nuanced, low-resource tasks [7]—we propose a Language-Specific Feature Selection pipeline. Our methodology avoids imposing rigid priors; instead, we engineer a broad “toolbox” of 61 sociolinguistic features and utilize a Random Forest ranking algorithm to empirically derive the most predictive drivers for each language.

Our contribution is twofold, bridging the gap between linguistic theory and computational implementation:

- **Linguistic Discovery:** By allowing the feature selection process to operate independently per language, we identified distinct mechanisms of reclamation. Our analysis reveals that English reclamation is primarily identity-driven (reliant on self-referential pronouns), whereas Italian reclamation is complexity-driven (characterized by syntactic density).
- **Computational Architecture:** We implement a dual-stream fusion model that integrates these empirically selected features with monolingual BERT embeddings. By processing a dense, language-optimized feature vector alongside contextual embeddings, our system effectively disentangles toxic slurs from empowering speech.

The proposed architecture delivers competitive classification results, yielding Macro F1-scores of 0.8981 for Italian and 0.6960 for Spanish—a resilient performance for the latter given the higher dialectal variance and data scarcity of the Spanish subset. These results validate the efficacy of a “Shared Toolbox, Different Tools” approach, where the architecture remains consistent, but the sociolinguistic lens adapts to the cultural reality of each language.

Warning: This paper contains examples of explicitly offensive content.

2. Dataset and Preprocessing

2.1. Dataset and Resources

The experimental data was provided by the MultiPRIDE @ EVALITA 2026 shared task organizers. To ensure proper attribution of the aggregated resources, we acknowledge the primary collections from which the monolingual sub-corpora were derived or adapted. The Italian data is primarily drawn from the TWITA collection [8], while the Spanish subset originates from the LGBTQI+ Dataset 2020-2022 [9]. For English, we note that the annotation schema and linguistic definition of reclamation align with the theoretical frameworks proposed by Zsisku et al. [10], focusing on the disambiguation of empowering usage from discrimination.

- **Label 1 (Reclaimed):** The use of potentially offensive terms for self-empowerment or community bonding (e.g., “I am a faggot and proud”).
- **Label 0 (Non-Reclamatory):** This category encompasses not only hate speech and toxic attacks but also neutral usage, and instances where the term is used without clear empowering intent.

The dataset was constructed by the organizers using the FATA (First Ask Then Act) annotation protocol, which utilizes community-based annotators to capture the “insider” pragmatic nuance often missed by standard crowd-workers [6].

2.2. Class Imbalance and Training Statistics

A comprehensive analysis of the training data revealed severe class imbalance, confirming that reclamation is a relatively rare phenomenon even within LGBTQ-focused discourse. The English subset proved most challenging, presenting the highest volume of data but the lowest density of positive instances. The specific distribution of the training set used for our feature engineering pipeline is as follows:

- **English:** 1,026 total samples, containing only 88 positive instances (8.5%). This extreme sparsity (approx. 1:11 ratio) poses the most significant challenge for the model.
- **Spanish:** 876 total samples with 133 positive instances (15.1%), representing a moderate imbalance (approx. 1:6.5 ratio).
- **Italian:** 1,086 total samples with 207 positive instances (19.0%), making it the most balanced subset (approx. 1:5.2 ratio), though still heavily skewed toward the non-reclaimed class.

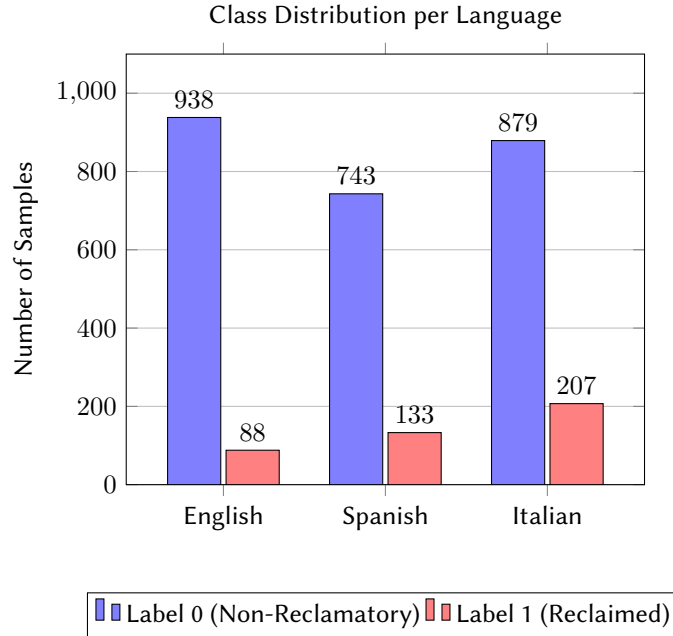


Figure 1: Distribution of Reclamatory (Label 1) vs. Non-Reclamatory (Label 0) instances across languages. The severe sparsity in the English subset (8.5% positive class) highlights the challenge of the task compared to Italian (19.0%).

2.3. Linguistic Feature Extraction

Our methodology prioritizes linguistic retention over aggressive cleaning. Unlike standard pipelines that strip punctuation or “noise,” we posit that these elements contain vital signals of tone and intent. Consequently, raw text was fed directly into spaCy Large pipelines (en_core_web_lg, es_core_news_lg, it_core_news_lg) without prior regex filtration for URLs or special characters. This approach allowed us to extract robust syntactic dependencies and Part-of-Speech (POS) tags, generating a feature space that captures structural complexity—a key hypothesis for our Italian model—without losing the “messy” context often found in social media text.

Table 1

Top 5 most predictive features per language identified by Random Forest. Note the prominence of structural features (length/counts) for Italian versus personal/sentiment markers (first_person_ratio) for English.

Rank	English Feature (Identity)	Imp.	Spanish Feature (Hybrid)	Imp.	Italian Feature (Complexity)	Imp.
1	first_person_ratio	0.0456	hate_speech_score	0.0597	text_length	0.1139
2	vader_compound	0.0384	hashtag_count	0.0514	char_count	0.1129
3	caps_ratio	0.0375	char_count	0.0508	hate_speech_score	0.1128
4	self_reference_ratio	0.0364	text_length	0.0506	word_count	0.0678
5	pos_ADP	0.0321	lgbtq_term_density	0.0421	sentiment_hate_gap	0.0518

2.4. Feature-Aware Oversampling

To prevent the Random Forest feature selection algorithm from overfitting to the majority class (Non-Reclaimed), we implemented a Targeted Oversampling strategy within the training loop. Rather than simple duplication, we utilized a dynamic sampling approach with a target ratio of 3.0 (reducing the imbalance to 1 minority sample for every 3 majority samples). This specific threshold was chosen to provide the model with sufficient exposure to the minority class patterns (Label 1) without creating an artificial 50/50 balance that ignores the real-world rarity of the phenomenon.

2.5. Task Subjectivity and Theoretical Limits

To contextualize the difficulty of the classification task, it is necessary to acknowledge the inherent subjectivity of defining “reclamation.” The use of the FATA protocol [6] by the organizers highlights that ground-truth labels are not objective facts but community-derived consensuses. This subjectivity introduces a “theoretical upper bound” on model performance, particularly evident in the English subset. Qualitative analysis suggests that English reclamation—often reliant on subtle identity markers rather than explicit lexical changes—suffers from higher annotator ambiguity than Italian, where reclamation is frequently marked by distinct structural complexity. This variance in human consensus correlates directly with our model’s downstream performance:

- **Italian:** The stability of linguistic markers (complexity/length) aligned with higher model confidence, yielding a robust Macro F1 of 0.8981.
- **English:** The high fluidity of the boundary between “slur” and “reclaimed term” resulted in significant class overlap, explaining the model’s struggle to exceed a Macro F1 of 0.5460.

Consequently, the errors in the English subset should be viewed not merely as architectural failures, but as a reflection of the sociolinguistic ambiguity inherent to the dataset itself.

3. Methodology

We propose a Hybrid Fusion Architecture that operates on the premise that sociolinguistic context is not fully captured by standard pre-trained language models. Our system decouples linguistic feature extraction from semantic representation, merging them via a deep fusion mechanism.

3.1. System Architecture

The model utilizes a dual-stream neural network structure processing the input text T through two parallel pathways:

Stream A: Contextual Semantic Embedding The raw text is processed by a monolingual BERT encoder to capture deep semantic dependencies. We utilized distinct checkpoints for each language (e.g., `bert-base-uncased` for English, `dccuchile/bert-base-spanish-wwm-uncased` for Spanish, and `dbmdz/bert-base-italian-uncased` for Italian) to ensure cultural alignment. We extract the pooled output or the mean of the last hidden state, yielding a semantic vector $h_{BERT} \in \mathbb{R}^{768}$.

Stream B: Sociolinguistic Feature Vector Simultaneously, the text undergoes the feature extraction pipeline producing a normalized vector $x_{ling} \in \mathbb{R}^{50}$. This vector is processed by a Feature Encoder MLP designed to project the explicit features into a latent space. The encoder consists of a linear projection to 128 dimensions, followed by Layer Normalization, ReLU activation, and Dropout ($p = 0.5$), yielding a sociolinguistic representation $h_{MLP} \in \mathbb{R}^{128}$.

Deep Fusion Mechanism Unlike simple concatenation, we implement a Deep Fusion Block to model the non-linear interactions between semantic and linguistic signals. The representations are concatenated ($v_{joint} \in \mathbb{R}^{896}$) and passed through a fusion network:

$$h_{fused} = \text{LayerNorm}(\text{ReLU}(W_f \cdot v_{joint})) \quad (1)$$

where the fusion layer projects the dimension down to 256. The final classification is performed via a linear head on this fused vector.

3.2. Linguistic Feature Engineering

We implemented a “Shared Toolbox” approach, extracting a superset of 61 candidate features using **spaCy Large** pipelines and the `twitter-xlm-roberta-base-sentiment` model. The features span three sociolinguistic categories:

- **Morphosyntactic:** Syntactic depth, Part-of-Speech densities, and sentence complexity metrics.
- **Sentiment Dynamics:** The “Sentiment-Hate Gap,” serving as a proxy for irony or reclamation.
- **Identity Markers:** Frequencies of self-referential pronouns (1st person) vs. other-referential pronouns.

3.2.1. Derived Sociolinguistic Metrics

To operationalize the “reclamation paradox”—where the semantic toxicity of a term contradicts its pragmatic intent—we engineered composite features that quantify the dissonance between conflicting linguistic signals. Let $S_{raw} \in [-1, 1]$ be the sentiment score derived from XLM-RoBERTa, which we normalize to $S = \frac{S_{raw}+1}{2} \in [0, 1]$. Let $T \in [0, 1]$ be the hate speech probability, R_{self} be the self-reference ratio (1st vs. 2nd/3rd person pronouns), and $E \in \mathbb{N}$ be the count of empowerment keywords. We defined the following heuristic metrics:

1. **Paradox Score (P):** Quantifies the coexistence of positive sentiment and low toxicity, a hallmark of successful reclamation. This feature penalizes toxic positive statements (e.g., sarcasm) by weighting sentiment against the non-toxicity probability.

$$P = S \times (1 - T) \quad (2)$$

2. **Sentiment-Hate Gap (G):** Measures the explicit mathematical distance between the model’s perceived sentiment and its toxicity detection. High gaps ($S \approx 1, T \approx 0$) serve as a proxy for the reappropriative shift.

$$G = S - T \quad (3)$$

3. **Reclamatory Intent Score (I_{rec}):** A composite heuristic that weights the Paradox Score (P) with explicit identity (R_{self}) and empowerment markers (E). This scalar value acts as a high-level indicator of “pride” usage.

$$I_{rec} = \frac{P \times (1 + R_{self}) \times (1 + E)}{4} \quad (4)$$

4. **Linguistic Complexity (L_c):** To test the hypothesis that reclamation in Romance languages involves higher syntactic elaboration, we derived a complexity score based on average word length (W_μ) and the unique word ratio (UWR).

$$L_c = W_\mu \times \text{UWR} \quad (5)$$

3.3. Model Complexity and Reproducibility

The total trainable parameter count for the Hybrid Fusion architecture is approximately 110.2 million. The majority of these parameters belong to the fine-tuned BERT backbone (approx. 109M), while the language-specific feature encoders and fusion layers introduce a lightweight overhead of fewer than 300,000 parameters. All models were trained on a single NVIDIA GeForce RTX 5090 GPU (32GB) using a batch size of 32 and the AdamW optimizer.

4. Results

4.1. Performance Analysis: The Explicit-Implicit Divide

Table 2 presents the classification performance of the Hybrid Fusion architecture. The results reveal a sharp divergence in model efficacy, directly correlated with the specific “drivers” of reclamation in each language.

While the model achieved robust performance in Italian (Macro F1: 0.8981) and moderate success in Spanish (Macro F1: 0.6960), it struggled significantly with English (Macro F1: 0.5460). Crucially, the Class 1 (Reclaimed) Recall for English was only 0.10. This indicates that while the model could identify non-reclaimed tweets with high precision, it failed to identify 90% of actual reclamation instances.

This disparity provides empirical evidence for our core hypothesis: Sociolinguistic features (e.g., pronouns, sentiment, sentence length) are strong predictors for explicit/argumentative reclamation (Italian) but fail to capture the implicit/contextual nature of English reclamation.

Table 2

Classification Performance and Primary Drivers

Language	Macro F1	Class 1 (Reclaimed) F1	Class 1 Recall	Primary Feature Driver (Rank #1)
Italian	0.8981	0.8333	0.7914	text_length (Complexity)
Spanish	0.6960	0.4638	0.3636	hate_speech_score (Toxicity)
English	0.5460	0.1463	0.1017	first_person_ratio (Identity)

4.2. Feature Importance and Sociolinguistic Drivers

To understand the mechanics behind these performance differences, we analyzed the Feature Importance rankings generated by the Random Forest selector. This analysis exposes the specific linguistic strategies the model attempted to leverage for each language.

4.2.1. Italian: The “Manifesto” Pattern (Complexity + Low Toxicity)

Italian reclamation was the easiest for the model to detect because it follows a distinct structural pattern.

- **Top Features:** text_length (Rank 1) and char_count (Rank 2) were the dominant predictors, followed closely by hate_speech_score (Rank 3) and sentiment_hate_gap (Rank 5).
- **Analysis:** The strong predictive power of the Sentiment-Hate Gap (Rank 5) contradicts the assumption that text length dilutes sentiment signals. Instead, the model learned a specific profile: High Complexity + Low Toxicity = Reclamation.

This suggests that Italian users reclaim slurs through “Manifestos”—long, well-reasoned, non-toxic arguments defending the community. The explicit nature of this style makes it highly detectable by feature-based models.

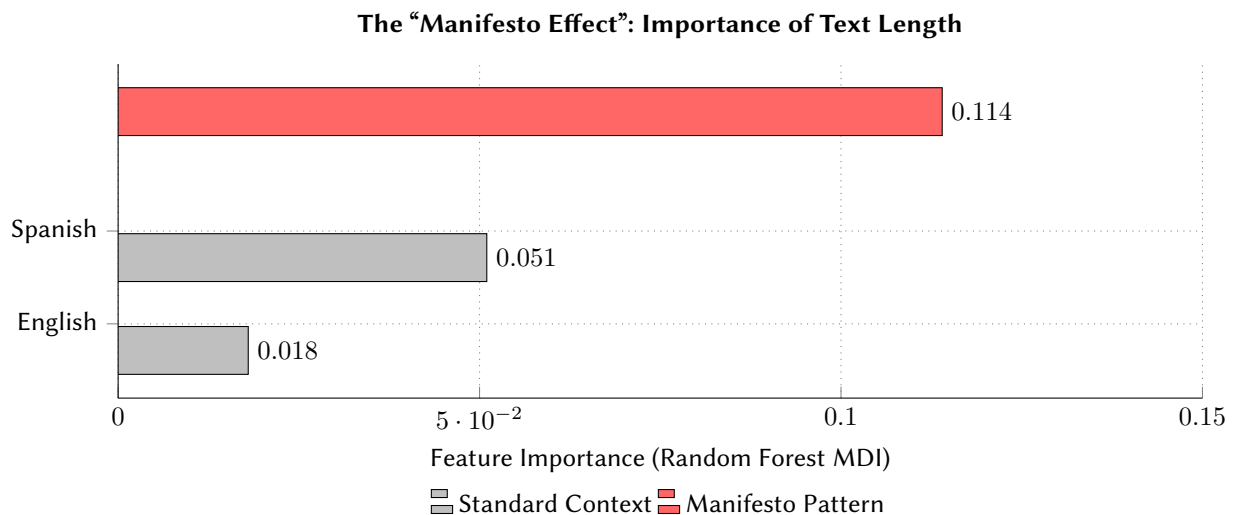


Figure 2: Feature Importance of “Text Length” across Languages. The Italian model (red) relies heavily on length (0.114), validating the “Manifesto Theory” of argumentative reclamation. In contrast, English and Spanish (gray) show significantly lower reliance on length, indicating implicit or toxicity-based drivers.

4.2.2. English: The Anomaly of Feature Poverty

In English, the Random Forest selected `first_person_ratio` as the primary feature (Rank #1), yet this selection coincided with a Recall of only 0.10 for the Reclaimed class. This discrepancy reveals a critical limitation: identity markers are necessary but not sufficient. While explicit reclamation often involves self-reference (e.g., “I am a faggot and proud”), the vast majority of non-reclaimed hate speech reporting also involves self-reference (e.g., “I got punched in the neck and called faggot in junior year of high school”). The model’s reliance on this feature indicates a state of “feature poverty” in English—it lacked the strong structural signals found in Italian (Feature Importance 0.114) and defaulted to weak identity signals (Feature Importance 0.046). Consequently, the model could not confidently distinguish Agents (reclaimers) from Objects (victims), resulting in a high false negative rate where genuine reclamation was lost in the noise of victimization reporting.

- **Top Features:** `first_person_ratio` (Rank 1), `vader_compound` (Rank 2), and `self_reference_ratio` (Rank 4).
- **Analysis:** The model prioritized who is speaking (using pronouns like “I”, “me”, “my”) over what was said. However, the catastrophic Class 1 Recall (0.10) proves that identity markers are a poor proxy for reclamation in English.

English reclamation often occurs through irony, in-group slang, or reappropriation directed at others (e.g., “You guys are queer icons”), which lacks first-person markers. By over-relying on “I am” statements, the model missed the vast majority of implicit reclamation, highlighting the limitation of surface-level sociolinguistic features for English.

4.2.3. Spanish: The Toxicity Threshold

Spanish occupied a middle ground, driven by explicit toxicity signals.

- **Top Features:** `hate_speech_score` (Rank 1) and `lgbtq_term_density`.
- **Analysis:** Unlike Italian (driven by structure) or English (driven by pronouns), Spanish reclamation was best predicted by the absence of hate speech signals in the presence of community terms. The model effectively acted as a “toxicity filter,” identifying reclamation as instances where slurs appeared without the accompanying venom of hate speech.

4.3. Error Analysis

To investigate the limitations of the Hybrid Fusion architecture, we conducted a qualitative analysis of misclassified samples (20 per language). This analysis reveals that the “drivers” identified in Section 4.2—while generally effective—create specific “blind spots” where the selected features conflict with the semantic reality of the text.

Table 3 summarizes the four primary failure modes identified across the languages.

Table 3

Feature Conflict Analysis in Misclassified Samples

Failure Mode	Lang	Example Text (Snippet)	The “Trap” (Feature that Fooled Model)	The “Truth” (Context Model Missed)
Agentivity Confusion	EN	“The one and only time someone called me a faggot... I punched him...” (en_299)	First-Person Pronouns: Model equates “me/I” + “slur” with identity performance.	Semantic Role: User is the Object (victim) of the slur, not the Agent (reclaimer).
Toxicity Barrier	EN	“Proud faggot reporting in...” (en_816)	Lexical Toxicity: The high negative weight of “faggot” in BERT embeddings.	Modifier Ignored: The explicit positive adjective (“Proud”) failed to override the slur’s toxicity.
Structural Mimicry	IT	“Insulti che vanno da finocchio a checca isterica...” (it_567) (<i>Insults ranging from faggot to hysterical sissy...</i>)	Complexity: Long, well-punctuated text (Rank #1 feature for IT).	Genre Confusion: User is reporting hate speech in a formal style, not arguing for reclamation.
Hierarchy Collapse	ES	“Maricones ¿vuestras amigas hetero están compartiendo... #OrgulloLGTBI” (es_423) (<i>Faggots, are your hetero friends sharing... #LGBTPrize</i>)	Positional Toxicity: Aggressive vocative at the start of the sentence.	Signal Hierarchy: The “softening” hashtag at the end was outweighed by the initial slur.

4.3.1. English: The Semantic Role Failure

The English model’s heavy reliance on `first_person_ratio` (Rank #1) creates a critical flaw we term “Agentivity Confusion.” The model acts on the heuristic that Slur + Self-Reference = Reclamation.

- **Case Study (en_299):** The theoretical failure of identity markers described in Section 4.2.2 is visibly demonstrated in sample en_299. The user writes: “The one and only time someone called me a faggot... I punched him.” This is a report of violence and victimization. However, the high density of first-person markers (“me”, “I”, “we”) triggered a False Positive prediction of Reclamation.
- **Implication:** This proves that statistical feature engineering cannot substitute for Semantic Role Labeling (SRL). The model correctly identified the presence of identity markers but failed to distinguish the user as the victim (object) rather than the agent (subject) of the slur.

4.3.2. Spanish: The Signal Hierarchy Conflict

In Spanish, we observed a conflict between the “Contextual Cushioning” features (hashtags) and the “Toxicity” features.

- **Case Study (es_423):** The tweet contains the explicit softening hashtag #OrgulloLGTBI. However, the text opens with an aggressive vocative: “Maricones...” (“Faggots...”). The model classified this as Hate Speech (False Negative).
- **Implication:** This contradicts the assumption that hashtags always act as “safeguards.” It suggests a Decision Hierarchy within the model: the raw toxicity of a slur positioned at the start of a

sentence (high saliency) overrides the contextual signal of a hashtag at the end. Conversely, in sarcastic tweets like es_372: “*Feliz día de la mariconeria #LGTBI*” (Happy day of faggotry #LGBT), the model over-trusted the hashtag, leading to a False Positive.

4.3.3. Italian: Genre Confusion

The Italian model’s success with `text_length` (F1: 0.8981) led to errors of “Structural Mimicry.”

- **Case Study (it_567):** The user provides a detailed list of insults they have received (“from faggot to hysterical sissy”). The text is long, complex, and grammatically standard—matching the “Manifesto” profile the model learned to associate with reclamation.
- **Implication:** The model effectively learned to detect a genre (formal argumentative discourse) but struggles when that same genre is used to report hate speech rather than perform reclamation.

5. Discussion

Our results challenge the conventional reliance in multilingual NLP on uniform representations that implicitly model sociolinguistic phenomena as consistent across languages [3, 7]. While massive models rely on shared semantic representations [7], prior work indicates that this alignment breaks down for high-context pragmatic tasks like hate speech detection [5, 4]. By decoupling feature selection, our error analysis confirms that reclamation is not a universal linguistic act, but a culturally distinct performance that resists a single feature set.

5.1. Italian: Reclamation as Argumentative Discourse

The superior performance of the Italian model (F1: 0.8981) was driven principally by syntactic complexity and text length. We posit that this reflects the specific nature of the Italian dataset, where reclamation frequently manifests as “Counter-Speech.”

Unlike casual slang, these instances are characterized by well-structured arguments, correct punctuation, and extended length. The model effectively learned to distinguish “effortful” defense of the community (Reclamation) from “low-effort” slurs (Hate Speech). This supports a “Manifesto Theory” of reclamation in Italian: users reclaiming slurs tend to do so within the context of formal, political argumentation, making the phenomenon structurally distinct and highly detectable.

5.2. English: The Agentivity Trap and Feature Poverty

Our hypothesis that English is “identity-driven” requires nuance. While the model heavily weighted identity markers (pronouns) as the primary feature (Rank #1), the resulting class-specific F1-score (0.15) demonstrates that this strategy acts as a trap. Unlike Italian, where *how* a statement is constructed (syntactic complexity) signals intent, English reclamation often hinges on semantic roles—specifically, distinguishing *who* does *what* to *whom*.

The error analysis reveals that standard sociolinguistic features cannot effectively distinguish the **Agent** (“I am reclaiming this term”) from the **Object** (“This term was used against me”). Lacking access to deep semantic role labeling (SRL), the model failed to resolve this “Agentivity Ambiguity.” It effectively collapsed into a conservative majority-class classifier, predicting “Non-Reclaimed” in nearly 90% of cases simply because surface-level identity markers were too ambiguous to serve as a reliable decision boundary. This finding suggests that future work in English reclamation detection must move beyond surface features to incorporate dependency-aware role labeling.

5.3. Spanish: Contextual Cushioning and Architectural Conflict

The intermediate performance of the Spanish model (Macro F1: 0.6960) positions it as a true sociolinguistic hybrid, bridging the structural density of Italian and the reliance on specific markers seen in

English. While the Italian model was driven by “Manifesto” style complexity, the feature analysis for Spanish reveals a multifaceted strategy we term *Contextual Cushioning*.

The dominant predictor, `hate_speech_score` (Rank 1), confirms that the model primarily functions as a “Toxicity Filter,” assessing whether the semantic weight of the text aligns with hate speech. Crucially, however, the subsequent feature rankings expose a dual-dependency absent in the other languages. The model relies on explicit Community Markers (`hashtag_count` at Rank 2, `lgbtq_term_density` at Rank 5) supported by moderate Structural Complexity (`char_count` at Rank 3). This suggests that Spanish reclamation is identified not strictly by argumentation (as in Italian), but by the presence of “safe” metadata (hashtags) that cushion the toxicity of the slur.

However, this cushioning is subject to a strict Signal Hierarchy; as seen in error analysis (Table 3), the lexical saliency of a sentence-initial slur often overrides the mitigating weight of trailing metadata, leading to False Negatives.

5.4. The Necessity of Language-Specific Drivers

Our findings demonstrate the risk of “zero-shot” or “joint-training” approaches in sociolinguistics. A model trained primarily on English patterns (looking for identity pronouns) would have failed completely on Italian (which looks for length and argumentation).

Spanish served as a hybrid case, where the Paradox Score (the divergence between lexical toxicity and semantic sentiment) proved most effective. This confirms that while the intent (reclamation) is shared, the mechanism differs: Italian uses Structure, English uses Context, and Spanish uses Sentiment/Toxicity contrast.

5.5. Limitations

The primary limitation of this study is the extreme class imbalance in the English dataset (8.5% positive class), which exacerbated the model’s bias toward the majority class. Additionally, our “Manifesto Theory” for Italian may be specific to the Twitter/X platform dynamics in Italy during the data collection period and may not generalize to other platforms like TikTok or Instagram.

6. Conclusion

This study presented a Hybrid Fusion Architecture for the detection of reclaimed slurs in multilingual social media data. By integrating pre-trained BERT embeddings with language-specific sociolinguistic feature vectors, we empirically demonstrated that the linguistic realization of reclamation is not universal, but culturally distinct.

Our results validate the “Manifesto Theory” for Italian, where reclamation manifests as high-complexity, argumentative discourse identifiable by structural features (F1: 0.8981). Spanish results (F1: 0.6960) positioned the language as a hybrid case driven by “Contextual Cushioning” (e.g., hashtags), though error analysis revealed a strict Signal Hierarchy where initial lexical toxicity often overrides explicit positive markers. In contrast, the performance gap in English (F1: 0.5460, Recall: 0.1017) exposes the limitations of explicit feature engineering for high-context pragmatics. We found that while English speakers often reclaim slurs through implicit irony and in-group signaling, current models—driven by identity markers like first-person pronouns—fail to distinguish these acts from hate speech reporting.

6.1. Future Directions

The failure to capture implicit reclamation in English suggests that text-only models have reached a performance ceiling. Future work should prioritize two specific technical advancements:

- **Semantic Role Labeling (SRL):** To resolve the “Agentivity Confusion” (Subject vs. Object) identified in our error analysis, explicit modeling of semantic roles is required to distinguish the agent of reclamation from the victim of hate speech.

- **Contextual Metadata:** Integrating user-level bio descriptions and social graph embeddings to provide the necessary disambiguation context when the text itself is implicit (e.g., “You guys are queer icons”).

We conclude that effective sociolinguistic modeling requires moving beyond “Universal Multilingualism” toward architectures that respect the specific structural, hierarchical, and pragmatic drivers of each language.

Declaration on Generative AI

(by using the activity taxonomy in ceur-ws.org/genai-tax.html):

During the preparation of this work, the authors used ChatGPT in order to: check grammar and refine the language style. After using these tool, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] Z. Waseem, T. Davidson, D. Warmusley, I. Weber, Understanding abuse: A typology of abusive language detection subtasks, in: Proceedings of the First Workshop on Abusive Language Online, Association for Computational Linguistics, 2017, pp. 78–84.
- [2] M. Dynel, The reappropriation of slurs: A corpus-based study of “slut” and “bitch” on reddit, *Journal of Pragmatics* 173 (2021) 162–177.
- [3] D. Hershcovich, S. Frank, H. Lent, M. de Lhoneux, M. Abdou, S. Brandl, E. Bugliarello, L. C. Piqueras, I. Chalkidis, R. Cui, et al., Challenges and strategies in cross-cultural nlp, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 6997–7013.
- [4] A. Lauscher, V. Ravishankar, I. Vulic, G. Glavas, From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 4491–4503.
- [5] D. Nozza, Exposing the limits of zero-shot cross-lingual hate speech detection, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), 2021, pp. 907–914.
- [6] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [7] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.
- [8] V. Basile, M. Lai, M. Sanguinetti, et al., Long-term social media data collection at the university of turin, in: Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR-WS, 2018, pp. 1–6.
- [9] J. Mata, E. Gualda, A dataset of Spanish tweets on people and communities LGBTQI+ during the COVID-19 pandemic 2020-2022, 2022. doi:10.5281/zenodo.14878434, zenodo Dataset.
- [10] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: Proceedings of the 16th ACM Web Science Conference, ACM, 2024, pp. 241–249.

GRUPPETTOZZO at MultiPRIDE: Detecting LGBTQ+ Reclamatory Intent via Context-Aware Transformers

Federico Traina^{1,†}, Alessandro Santoro^{1,†}, Gabriele Greco^{1,†}, Irene Siragusa^{1,*} and Roberto Pirrone¹

¹Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 6, Palermo, 90128, Sicily, Italy

Abstract

In this report, we present the proposed approach of the GRUPPETTOZZO team for the detection of reclamatory usage of terms related to LGBTQ+. In the context of the multilingual MultiPRIDE challenge in the EVALITA 2026 campaign, we proposed a transformer-based encoder-only method to classify the textual content of the given tweets (task A) and in conjunction with the contextual information provided by the user biography (task B). To handle the unbalance in the given data set, we explored the usage of both weighted and focal loss functions and two data augmentation strategies, to better generalize the performance of developed models in the target multilingual context. We submitted two runs for each task, one using a target monolingual Language Model for each specific language, and one using a multilingual one. Both proposed approaches, overcame the baseline 6 times out of 10 and reached the 1st place in subtask B2.

Keywords

Context-Aware Transformers, LGBTQ+ Slurs Reclamation, Data Augmentation, Language Models

1. Warning

This paper contains examples of explicitly offensive content.

2. Introduction

Language within the LGBTQ+ context is characterized by complex phenomena such as reclamation, whereby terms originally intended as slurs are repurposed by community members for positive and/or identity-affirming purposes. In the context of social networks and hate speech detection applications, an important challenge lies in distinguishing the correct usage of such slurs that may have both a reclamation and a denigratory intent. MultiPRIDE challenge [1] attempts to address this complex phenomenon proposing two multilingual binary classification tasks. In particular, the objective is to determine whether a given tweet, uses a term related to LGBTQ+ context with a reclamatory intent (task A) in Italian (subtask A1), Spanish (subtask A2) and English (subtask A3). In addition, organizers add the user biography (user bio) when available as extra context for the classification (task B) in Italian (subtask B1) and Spanish (subtask B2), as well as encouraging cross-lingual applications (subtask A-multi and B-multi).

This paper describes the system developed by the GRUPPETTOZZO team for tasks A and B of the MultiPRIDE challenge in the context of the EVALITA 2026 campaign [2]. Our approach consists in fine-tuning encorded-only Language Models (LMs) to address task A and, for task B, we integrated textual biographies as pragmatic context into the model's input. Moreover, we handle tasks A and B in both monolingual and multilingual configuration. Our approach can be summarized in five stages,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ federico.traina01@community.unipa.it (F. Traina); alessandro.santoro04@community.unipa.it (A. Santoro); gabriele.greco07@community.unipa.it (G. Greco); irene.siragusa02@unipa.it (I. Siragusa); roberto.pirrone@unipa.it (R. Pirrone)

ORCID 0009-0005-8434-8729 (I. Siragusa); 0000-0001-9453-510X (R. Pirrone)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

as can be seen in Figure 1. Firstly we perform a data augmentation phase to enrich the original data set, following two different strategies. Textual contents are pre-processed through a data cleaning procedure before being injected into the developed Context-Aware Architecture, made up of a LM backbone, a custom pooling strategy and a classification head that performs the final prediction.

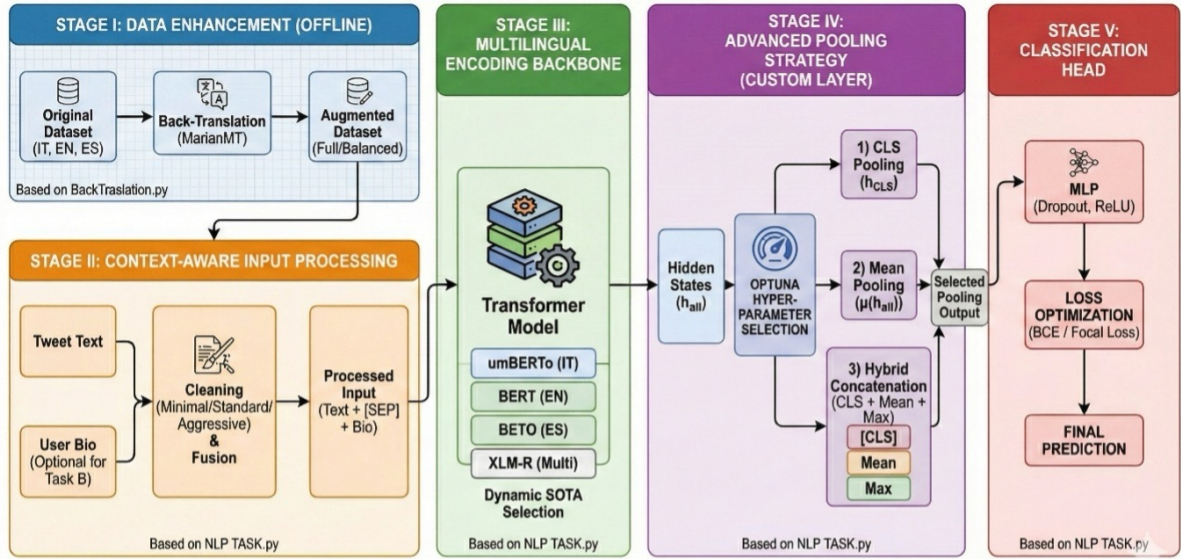


Figure 1: Overview of the proposed approach.

We submitted two runs for each subtask using a monolingual and a multilingual LM as backbone. The architecture developed was able to exceed the baseline 6 times out of 10, and reached the 1st place in subtask B2.

This report is structured as follows: data augmentation and pre-processing procedures, and the proposed system are described in Section 3, while results and related discussion are presented in Sections 4 and 5. Section 6 contains final remarks.

3. Description of the System

MultiPRIDE data set is arranged in three different splits, one for each language, comprehensive of 1,000 labeled tweets, where positive classes are associated to usage of slurs in a reclamatory manner. For Spanish and Italian splits, the user bio field was provided to address task B, if available. For the development phase, we split the data set using an 80-20 ratio and a stratified strategy.

The system processes data in two distinct structural modalities designed to evaluate the impact of external context on classification, depending exclusively on the target task and not on the language:

- **Text-Only** the input to the model is exclusively the textual content of the tweet (subtasks A1, A2, A3, A-multi);
- **Context-Aware** an enriched representation is generated by concatenating the original tweet with the user bio (subtasks B1, B2, B-multi). The input follows the pattern:

TWEET [SEP] CONTESTO: BIO

where the [SEP] token serves as a segment delimiter, thus enabling the model to distinguish between the two distinct information sources within the same input sequence. Whenever the user bio is not available, it is set to an empty string.

3.1. Data Augmentation and Pre-processing

Given the unbalanced distribution of positive labels and the limited number of samples in the training set, we implemented a back-translation pipeline based on MarianMT [3] and OPUS-MT [4, 5] models. This technique involves translating the text from the source language to a pivot language and subsequently re-translating it back into the original language. This process generates a paraphrase that preserves the core semantic content while introducing lexical and syntactic variance. We used English as pivot language for Italian and Spanish splits, while Spanish as pivot language for English split.

We experimented with two augmentation configurations as shown in Figure 2:

- **Full Augmentation** A paraphrase is generated for *every sample* in the data set. On doing this, we double the training set size, exposing the model to greater linguistic variability and helping to prevent overfitting on specific syntactic patterns.
- **Balanced Augmentation** Augmentation is applied *exclusively* to the minority class, i.e. the positive labeled samples. This technique reduces the class imbalance ratio, enabling the model to learn more robust features for the positive class without resorting to naive oversampling, which would simply duplicate existing examples.

After the data augmentation phases, a check over the generated samples have been conducted and eventually duplicated samples, which may result if the back-translation system outputs the original text, have been removed.

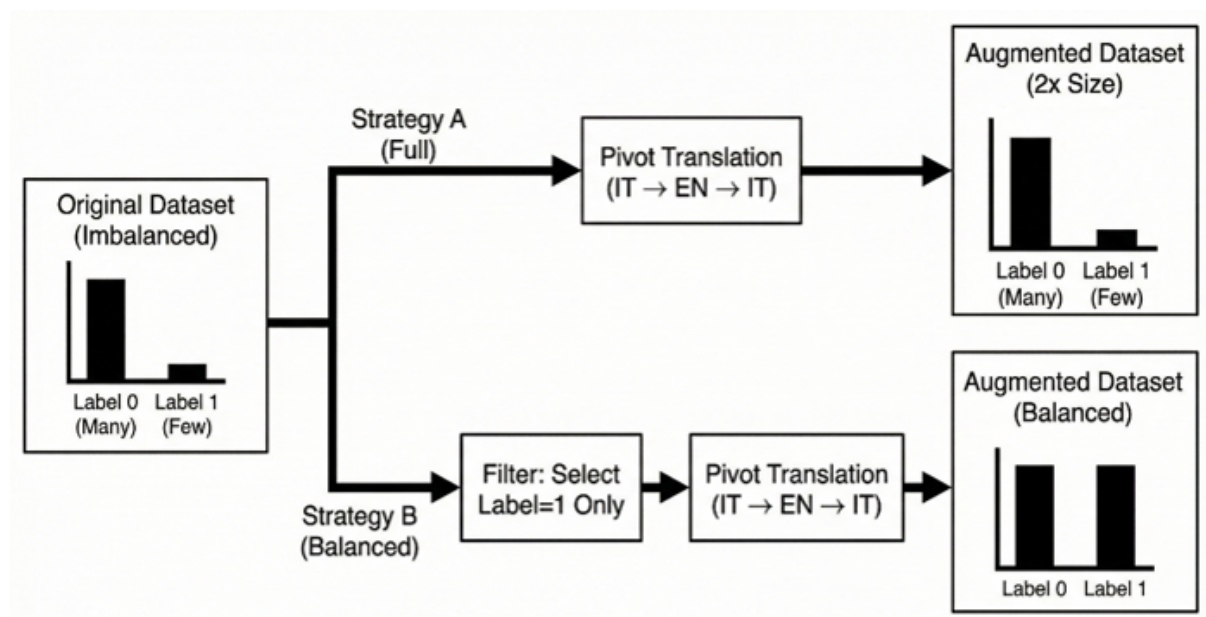


Figure 2: Overview of the data augmentation pipeline with full and balanced augmentation strategies.

Given the social media nature of the data set, we formalized three different strategies of data pre-processing, and we investigated the impact of each strategy:

- **Minimal** the original text is converted in lowercase and no further data cleaning operation are performed, thus the textual content is left unchanged and are kept both the total amount of information and any noisy information.
- **Standard** a demojization phase is performed over the lowercased input text, and any present emoji is converted in a textual string, e.g. 🌈 → :rainbow_flag:.
- **Aggressive** both emoji and stopwords are removed and textual content is lowercased.

The choice to lowercase the input in all the pre-processing strategies is a strategic decision for our task. On social media, capitalization is often irregular or used exclusively for emphasis. Normalizing

the text helps the model focus on semantic content rather than graphical variations, thus enhancing its generalization capabilities.

Emojis are dense semantic and sentiment information. Removing them eliminates any potential signals of LGBTQ+ community affiliation (e.g., colored hearts, pride flags) or references to the sentiment of the given tweet. The demojization projects these symbols into the semantic space of the BERT tokenizer [6], allowing the model to properly interpret those meta-linguistic features. In the aggressive strategy, the stopwords removal can be detrimental in pragmatic tasks, as functional particles (negations or pronouns) which can radically alter the sentence’s meaning.

3.2. Context-Aware Architecture

The proposed Context-Aware Architecture departs from standard classification models by integrating modular components designed to maximize semantic feature extraction and enhance model robustness in the presence of imbalanced data. It is made up of a LM backbone, followed by a neural module which further processes the dense representation obtained from the LM with a custom pooling strategy and a classification head that performs the final prediction.

3.2.1. Language Model Backbones

The proposed model leverages on fine-tuning of pre-trained transformer models [7], selected to maximize the semantic representation in each target language.

For the Italian split, we used `lupobricco/umBERTo_fine-tuned_hate_offensivity`¹ (umBERTo). This model has been obtained after a domain-specific fine-tuning of UmBERTo [8], which is an Italian LM based on RoBERTa model [9], and trained over a vast Italian corpus using the Whole Word Masking (WWM) technique. Specifically, the WWM requires the model to predict entire words even when only a sub-word unit is masked. This results to be a crucial feature in the context of Romance languages (Italian, Spanish), which are characterized by an abundance of suffixes and complex conjugations. In addition, UmBERTo more effectively handles the complex morphology of the Italian language by utilizing a SentencePiece-based tokenizer, which better preserves the semantic roots of words. The choice of umBERTo provides a critical advantage in Task A. This specific version has been further fine-tuned for Hate Speech Detection, thus resulting (i) already optimized to recognize patterns of toxicity and (ii) the need of an extensive training from scratch to identify aggressive intent is significantly reduced.

For the English split we adopted the standard BERT model in its base and uncased version [6], which represent a state-of-the-art LM.

As for the Spanish data set, we used BETO [10], a BERT-based model trained with the WWM technique over Spanish data. Analogously for the chosen Italian LM, WWM allows a better modeling of the Spanish language and the model itself is forced to learn deeper syntactic and semantic representations, consistently outperforming standard Multilingual BERT (mBERT) [6].

We also considered a multilingual backbone for the cross-lingual classification task. We opt for XLM-RoBERTa [11], which scales cross-lingual pre-training utilizing 2.5 TB of filtered CommonCrawl text across 100 languages. Unlike mBERT, XLM-RoBERTa omits the Next Sentence Prediction (NSP) objective and is trained with significantly larger batches and longer durations. This architecture facilitates the creation of a shared semantic space where semantically equivalent terms in different languages (e.g. ‘gay’ in English and ‘gay’ in Italian) are mapped to proximal vectors. This cross-lingual alignment capability is fundamental to our system, enabling the transfer of knowledge from high-resource examples in English and Spanish to improve predictions even for underrepresented classes [12]

3.2.2. Neural Module Architecture

Despite the different LM backbone used, the output of the last layer $H \in \mathbb{R}^{l \times d}$, where l is the length of the given sequence and d the dimension of the hidden state, serves as input to a neural model.

¹https://huggingface.co/lupobricco/umBERTo_fine-tuned_hate_offensivity

From this vector representation, three pooling strategies are defined:

- **CLS Pooling** $\mathbf{v}_{cls}$ which selects the embedding of the [CLS] token;
- **Mean Pooling** $\mathbf{v}_{mean}$ across the target sentence removing any padding tokens;
- **Hybrid Pooling** is the proposed advanced concatenation strategy, which consists in concatenating the vector representation obtained by the previous pooling strategies and a Max Pooling $\mathbf{v}_{max}$ over the dense representations for each sample, removing any padding token. The final representation is obtained as $\mathbf{v}_{final} = \mathbf{v}_{cls} \oplus \mathbf{v}_{mean} \oplus \mathbf{v}_{max}$, with dimension equal to $3 \times d = 2304$.

While $\mathbf{v}_{cls}$ is optimized during LM training, $\mathbf{v}_{mean}$ captures the global context, and $\mathbf{v}_{max}$ highlights salient signals (e.g., a single offensive term). The concatenation of these vectors provides a more comprehensive overview of the input data to the final classification.

The classification head applies, over the obtained data after pooling, a dropout layer to prevent the overfitting of the network, followed by a hidden dense layer, a ReLU activation function and the classification layer.

3.3. Training Details

For the training phase, we adopted an hyperparameter optimization strategy, including architectural details such as the pooling strategy, dropout value, the size of the hidden MLP layer, and training details as the number of layers to freeze in the LM backbone, loss function, optimizer and learning rate, described below.

To balance learning capacity with the stability of pre-trained weights in the LM backbones, we implemented a selective freezing mechanism. The initial embeddings and the firsts n encoder layers can be frozen in order to (i) force the model to leverage its acquired foundational linguistic knowledge and (ii) focus the training process exclusively on the higher-level layers. Such high-level features, then combined with the neural model architecture, can deeply focus toward the target task during its training phase.

We considered two different options as for the loss function: a weighted Binary Cross-Entropy (BCE) loss with the inverse class ratio as weight, and a Focal Loss [13], parametrized with γ , optimized to down-weight easy samples, and compelling the model to focus on more complex, hard-to-classify instances.

As optimizer during the training phase, we both evaluated AdamW [14] and EvoLved Sign Momentum (Lion) [15]. Unlike AdamW, which computes the first and second moments (mean and variance) of the gradients, the Lion optimizer utilizes only the sign of the gradient for weight updates. This approach simplifies the optimization process, reduces memory overhead, and provides a more robust regularization effect, specifically in fine-tuning on small data sets such as the MultiPRIDE one.

We used Optuna [16], a bayesian optimization to determine the optimal hyperparameter configuration for each task and subtask, and a comprehensive overview of all the considered hyperparameters reported Table 1.

Table 1

Overview of all the considered hyperparameters.

Hyperparameter	Search space / possible values
Pooling strategy	CLS Pooling, Mean Pooling, Hybrid Pooling
# frozen layers LM	1,2,3,4,5,6
MLP size	128, 256, 623
Dropout	from 0.1 to 0.5
Loss Function	Weighted BCE, Focal Loss
Gamma (Focal Loss)	from 0.5 to 3
Learning Rate	from 1e-5 to 5e-5
Optimizer	AdamW, Lion

The developed system and the training pipeline have been implemented in PyTorch [17] and with HuggingFace Transformers [18]. To optimize the computational efficiency we adopted a Mixed Precision Training (FP16) technique. We trained the models up to a maximum of 8 epochs and implemented an Early Stopping strategy over the macro F1-score in our validation set with a patience value equal to 3. Experiments have been conducted over Google Colab instances equipped with one GPU Tesla T4 and Kaggle instances with one NVIDIA TESLA P100 GPU.

4. Results

For each subtask diverse training configuration have been explored, namely the diverse augmentation strategies (without, full and balanced), data cleaning techniques (minimal, standard, aggressive), combined with the best hyperparameter obtained with OPTUNA for each configuration (Table 1). In this section we report the configurations which obtained the best results in terms of macro F1-score over our validation set and the official test set for both tasks.

In Table 2 are reported our development results for Task A, computed on an internal validation set obtained via a stratified 80/20 split of the official training data.

Table 2

Results over internal validation set for Task A. Bold results refer to the highest ones per language split.

Subtask	Data Cleaning	Data Augmentation	Pooling	# frozen LM layers	MLP size	Optimizer	Loss	Macro F1-score
A1 (IT)	Standard	None	CLS	2	128	Lion	BCE	0.9014
A2 (ES)	Minimal	None	Mean	1	128	AdamW	BCE	0.7320
A3 (EN)	Minimal	None	CLS	4	128	Lion	Focal	0.6288
A-Multi	Aggressive	None	Hybrid	5	128	AdamW	Focal	0.7898
A1 (IT)	Minimal	Full	Hybrid	4	128	Lion	BCE	0.9702
A2 (ES)	Minimal	Full	CLS	6	128	Lion	Focal	0.9047
A3 (EN)	Minimal	Full	Mean	4	128	AdamW	Focal	0.8849
A-Multi	Standard	Full	Mean	3	128	Lion	Focal	0.8816
A1 (IT)	Minimal	Balanced	Hybrid	0	128	Lion	Focal	0.9478
A2 (ES)	Minimal	Balanced	Mean	2	128	Lion	Focal	0.8987
A3 (EN)	Standard	Balanced	Hybrid	4	256	Lion	Focal	0.9190
A-Multi	Standard	Balanced	Mean	5	128	Lion	Focal	0.8871

Lowest results are obtained without data augmentation, suggesting an intrinsic complexity of the task, especially in contest of data scarcity. Both strategies of full and balanced data augmentation led to higher performances, almost over 0.90 for all configurations, confirming that the initial poor generalization capabilities are mainly addressed to the reduced number of training samples.

The Italian split (subtask A1) shows the overall best results, even without data augmentation and increasing with both full and balanced augmentation. This behavior may suggest an minor linguistic ambiguity in Italian tweets, when compared with other languages. While both Italian and Spanish data benefits from the full augmentation strategy, English and multilingual splits highly benefit from the balanced augmentation.

As for data cleaning, the minimal strategy, which essentially left the input data unchanged, resulted to be the overall best choice, thus assessing that punctuation elements and emojis are semantically essential for the given task. Multilingual setup, on the contrary, needs a standard or aggressive data pre-processing strategy. These performances show that a normalization step helps cross-lingual models to better map concepts in a common latent and embedding space.

Regarding the pooling strategies, results obtained with the concatenation of the three pooling operations (hybrid pooling) are the best one for Italian and English splits, while Spanish split benefits from the usage of the simple [CLS] token, while the mean pooling was preferred for the multilingual setup.

The conjunct use of the focal loss with Lion optimizer resulted overall the best choice to focus the learning towards the more complex samples and the unbalanced class. Furthermore, the optimal size for the MLP layer was equal to 128 and overall best results are obtained though a soft training of the LM backbone, that is with a higher lever of transformer layers kept frozen during the training.

In Table 4 are reported the obtained results for Task B computed on an internal validation set obtained via a stratified 80/20 split of the official training data, within the different augmentation strategies for each language and with the multilingual configuration. In this task, the additional information provided by the user bio, introduces identity elements about the author of the target tweet, which can used by the model as suggestion for the final classification.

Table 3

Results over validation set for Task B. Bold results refer to the highest ones per language split.

Subtask	Data Cleaning	Data Augmentation	Pooling	# frozen LM layers	MLP size	Optimizer	Loss	Macro F1-score
B1 (IT)	Aggressive	None	CLS	1	256	AdamW	Focal	0.9043
B2 (ES)	Standard	None	Hybrid	1	128	AdamW	Focal	0.7064
B-Multi	Aggressive	None	Hybrid	1	256	AdamW	Focal	0.8340
B1 (IT)	Standard	Full	Mean	4	128	AdamW	Focal	0.9667
B2 (ES)	Standard	Full	CLS	0	256	AdamW	Focal	0.9541
B-Multi	Standard	Full	Hybrid	3	128	Lion	BCE	0.9015
B1 (IT)	Standard	Balanced	Hybrid	3	128	Lion	BCE	0.9602
B2 (ES)	Aggressive	Balanced	Hybrid	5	128	AdamW	Focal	0.9010
B-Multi	Standard	Balanced	Hybrid	0	128	Lion	Focal	0.9041

Also for this case, the full augmentation results the best data augmentation option, with the only exception for the multilingual case, which slightly benefits from the balanced augmentation. Overall, performances of the developed models without data augmentation are considerably lower compared to the ones with data augmentation.

Despite the additional information provided by the user bio, this did not result sufficient to compensate both the unbalanced labels distribution and the poor quantity of samples, making them comparable to the results obtained with the task A. More specifically, Italian and Multilingual splits benefits from the additional information of the user bio, compared to the Spanish one, whose performances decrease. We hypothesize that the model can effectively benefits of this additional context for the cases in which a high coherence is present between the user bio and the target tweet, suggesting that in the Italian split this correlation is higher than in the Spanish one. However, the overall best performances are obtained with the full data augmentation strategy and implementing the standard data cleaning method and a training based on Focal Loss and AdamW as optimizer.

For each task, we submitted the prediction obtained with both monolingual (run 1) and multilingual (run 2) model reaching the best performances over our validation set (Tables 2 and 3), and in Table 4 are collected the obtained results over the test set.

Obtained results confirms the excellent performance of the fine-tuned umBERTo model. Specifically, in subtask B1 we obtained a macro F1-score of 0.8979 with the monolingual proposed architecture. On the other hand, the integration of user biographies did not lead to an improvement: the macro-F1 went from 0.8735 (Task A) to 0.8681 (Task B) in the proposed multilingual model.

A drastic performance decline was observed for the English split in subtask A3, compared to the result observed during the development phase, where we reached an macro F1-score of only 0.5979 with the multilingual backbone. This suggests either significant overfitting on the training data or a semantic distributional shift in the test set compared with the training one. The different nature of the data may led the model to struggled to differentiate between reclaimed usage and actual hate speech.

For Spanish subtasks, the monolingual system achieved a macro F1-score of 0.7289 in subtask A2 and

Table 4

Results over test set for Task A-B. Official score for ranking is the Macro F1-Score

Subtask	Run	Rank	Macro Precision	Macro Recall	Macro F1-Score
A1 (IT)	1	6	0.9144	0.8594	0.8834
A1 (IT)	2	9	0.8725	0.8745	0.8735
A2 (ES)	1	8	0.7387	0.7205	0.7289
A2 (ES)	2	5	0.7273	0.7882	0.7506
A3 (EN)	1	14	0.5516	0.5248	0.5285
A3 (EN)	2	6	0.5866	0.6184	0.5979
B1 (IT)	1	3	0.9061	0.8903	0.8979
B1 (IT)	2	6	0.9247	0.8313	0.8681
B2 (ES)	1	1	0.7293	0.7315	0.7304
B2 (ES)	2	8	0.7506	0.6560	0.6856

0.7304 in subtask B2, in which placed in the first spot of the ranking. This corresponds to an absolute gain of 0.0015 macro F1 points for A2 over B2. This suggests that, for the Spanish test set, biographical information may have introduced contextual noise rather than clarifying the intent of the speaker. Moreover, the usage of a monolingual model was crucial compared with a multilingual one, for which a severe downgrade in performances with the additional bio information can be seen.

5. Discussion

A deeper analysis of the results obtained allows us to better analyze the complexity of the task and compare the effectiveness of the proposed techniques.

A counterintuitive finding from our experiments is the superiority of less invasive data cleaning strategies (minimal and standard) compared to the aggressive one. In traditional NLP tasks, stopwords and punctuation are often discarded as noise. However, within the LGBTQ+ linguistic context, the use of emojis (e.g., 🏳️‍🌈, 🌟) and pronouns frequently serves as a crucial indicator of the author’s identity and its possible reclaimed intent. The removal of these elements, in the aggressive strategy, deprives the model of fundamental pragmatic signals.

The implementation of a augmentation strategy with Back-Translation proved to be pivotal. Models trained exclusively on the original data set exhibited clear signs of data scarcity and struggled with generalization. The introduction of augmented variants stabilized the training process, enabling the multilingual model (XLM-RoBERTa) to achieve performance levels comparable to those of monolingual models, reaching a macro F1-score higher than 0.90 in development phase in tasks B. This demonstrates a significant cross-lingual transfer learning capability of the model, where the model successfully leverages augmented patterns across the three languages.

An outperforming pooling strategy cannot be uniquely identified, but the overall highest performance in both development and testing phases are addressed to the proposed hybrid pooling. This suggests that the salient information is not merely concentrated in the [CLS] token of the sequence, but is distributed across the tweet and the biography (mean and max pooling).

Moreover, the usage of the focal loss during the training guaranteed a more robust and generalization capabilities in the model itself which performances did not deteriorated significantly during the testing phase, with the only exception for subtask A3.

5.1. Error Analysis

A detailed examination of misclassified instances reveals systematic patterns that illuminate the intrinsic challenges of reclamation detection across the three languages examined.

Misclassification patterns reveal distinct difficulties across the three languages. In Italian, the model frequently misinterprets meta-linguistic discussions about slur usage as reclamation attempts. For instance the following tweet

"Ah perché dire 'sei troppo femminile' è omofobia ma coglione, frocio e compagnia bella no?"

discusses the inconsistency of considering certain terms offensive while accepting others. Overall it results as a reflexive commentary on language norms rather than genuine reclamation. Similarly, expressions like

"Prima che le haters mi inizino a dire eH mA qUeStA è OmOfObIa, volevo comunicarvi che frocia lo sono pure io"

represent preemptive self-identification that the model struggles to contextualize correctly.

In Spanish, the presence of explicit LGBTQ+ pride hashtags (e.g., #OrgulloLGTBI, #LGTBI) creates ambiguity. The model tends to overpredict reclamation when these markers appear alongside slurs, even in non-reclamatory contexts. For example,

"Gracias @USER por #Mansos y por #MaricónPerdido. Gracias a las personas que no se callan. Gracias a las personas que escuchan. Gracias a las personas que leen. Cómo nos gusta leer con orgullo! Feliz día #OrgulloLGTBI #OrgulloLGTBIQ+ #Orgullo2021 "

expresses gratitude for LGBTQ+ literature (referencing book titles #Mansos and #MaricónPerdido) within a celebratory pride context. Despite the positive framing and absence of reclamatory intent, the co-occurrence of pride-related terminology and the slur "maricón" in the book title confounds the classifier.

English presents the most severe challenges, with the majority of errors involving subtle pragmatic distinctions. The data set contains numerous instances of meta-discourse about reclamation itself, such as

"I don't like the word faggot and don't like the word queer, either. I see them both as insults"

where community members explicitly reject reclamation. The model fail to correct detect this particular nuance, misclassify it as reclamation. Additionally, ironic or sarcastic usages like

"Honestly though faggot is such a fun word to say and all of my lgbt friends and I call each other that like 24/7"

are systematically misclassified as reclamatory due to the positive framing and in-group reference, despite lacking genuine empowerment intent.

Certain reclamatory strategies prove particularly challenging across all languages. Empowerment through defiance—exemplified by the English tweet

"I enjoy being the wrong faggot to fuck with. Plus, sometimes it's just fun to fight"

which was annotated as reclamatory, requires understanding implicit self-empowerment signals that extend beyond explicit pride markers. Similarly, the Italian expression

"BUONGIORNO AMICI BUON MESE DELLA FROCIAGGINE, LESBAGGINE, BISESSUALISMO, TRANSGENDERESIMO QUEER"

employs creative neologisms and celebratory tone that the model struggles to distinguish from hate speech.

The biography context, despite being incorporated into the model, provides insufficient disambiguation in many cases. Users who identify as LGBTQ+ in their bios may discuss homophobic incidents they experienced without reclaiming the slurs used against them, yet the model systematically misinterprets author identity as sufficient evidence for reclamation.

6. Conclusion

In this paper we reported the architecture proposed by the GRUPPETTOZZO team for MultiPRIDE tasks A and B promoted at the EVALITA 2026 campaign. Our experimental results demonstrate that the identification of reclamation intent in slurs' usage is not a mere lexical problem, but a semantic one. Developed system won in subtask B2 reaching a macro F1-score of 0.7304, while reaching competitive results in the other subtasks and the baseline 6 times out of 10. Obtained results shows that preserving the non-verbal pragmatic markers (emojis) while adopting a suitable data augmentation strategy and a training procedure based on a focal loss, were our core features to increment model performance over the two proposed tasks. The concatenation of the textual content and the user bio for tasks B, allowed a performance increment when compared to tasks A for both Italian and Spanish split when a monolingual LM was used as backbone. On the contrary, the opposite behavior is found for multilingual model, which struggles in discriminating the additional user bio information.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini 3 in order to: Text Translation, Grammar and spelling check, Paraphrase and reword. Further, the author(s) used Gemini 3 for figures 1, 2 and 3 in order to: Generate images. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] M. Junczys-Dowmunt, R. Grundkiewicz, T. Dwojak, H. Hoang, K. Heafield, T. Neckermann, et al., Marian: Fast neural machine translation in C++, in: Proceedings of ACL 2018, System Demonstrations, Melbourne, Australia, 2018, pp. 116–121. URL: <https://aclanthology.org/P18-4020/>. doi:10.18653/v1/P18-4020.
- [4] J. Tiedemann, M. Aulamo, D. Bakshandaevala, M. Boggia, S.-A. Grönroos, T. Nieminen, A. Raganato, Y. Scherrer, R. Vazquez, S. Virpioja, Democratizing neural machine translation with OPUS-MT, Language Resources and Evaluation (2023) 713–755. doi:10.1007/s10579-023-09704-w.
- [5] J. Tiedemann, S. Thottingal, OPUS-MT – Building open translation services for the World, in: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT), Lisbon, Portugal, 2020.
- [6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
- [8] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020.

- [9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
- [10] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.
- [11] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, et al., Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.
- [12] D. Nozza, Exposing the limits of zero-shot cross-lingual hate speech detection, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Association for Computational Linguistics, Online, 2021, pp. 907–914. URL: <https://aclanthology.org/2021.acl-short.114/>. doi:10.18653/v1/2021.acl-short.114.
- [13] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007. doi:10.1109/ICCV.2017.324.
- [14] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: <https://arxiv.org/abs/1711.05101>. arXiv:1711.05101.
- [15] X. Chen, C. Liang, D. Huang, E. Real, K. Wang, H. Pham, X. Dong, T. Luong, C.-J. Hsieh, Y. Lu, Q. V. Le, Symbolic discovery of optimization algorithms, in: Proceedings of the 37th International Conference on Neural Information Processing Systems, Curran Associates Inc., Red Hook, NY, USA, 2023.
- [16] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, New York, NY, USA, 2019, p. 2623–2631. URL: <https://doi.org/10.1145/3292500.3330701>. doi:10.1145/3292500.3330701.
- [17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, volume 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: <https://proceedings.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [18] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, 2020, pp. 38–45.

HateItOff at MultiPRIDE: Linguistic and Sentiment Cues in Reclaimed LGBTQ+ Slur Detection*

Greta Damo^{1,*†}, Nicolás Benjamín Ocampo^{2,*†}

¹Université Côte d'Azur, CNRS, Inria, I3S
Route des Colles 930, 06903 Sophia Antipolis, France

²Centrum Wiskunde & Informatica
Science Park 123, 1098 XG Amsterdam, The Netherlands

Abstract

Reclaimed slurs challenge automatic hate speech detection, as terms historically used against the LGBTQ+ community may be reappropriated with positive or self-identifying intent. Traditional lexicon-based approaches often fail to capture these nuances, leading to misclassification errors. To address this challenge, this paper presents our participation in the MultiPRIDE shared task at EVALITA 2026. We addressed both textual (Task A) and contextual (Task B) reclaimed language detection for Italian and Spanish. We explored transformer-based models enriched with sentiment analysis information and linguistic features. Our results demonstrate that these features improve performance over the baselines. In Task A, our system ranked 8th out of 24 submissions for Italian and 6th out of 19 submissions for Spanish. In Task B, we achieved 11th place out of 18 submissions for Italian and 5th place out of 12 submissions for Spanish. An error analysis highlights persistent challenges related to pragmatic ambiguity, in-group criticism, and meta-linguistic uses of slurs.

Keywords

Hate speech, Reclaimed slurs, Social Media

Content Warning: This paper contains examples of language which may be offensive to some readers.

1. Introduction

Social media has become a primary medium for communication and self-expression, enabling users to share opinions, identities, and experiences at an unprecedented scale. For the LGBTQ+ community, online platforms can represent spaces of visibility and empowerment, but also environments where language is contested and meaning is highly context-dependent. In particular, terms historically used as slurs may be reappropriated and reclaimed by community members, acquiring positive or self-identifying meanings that differ substantially from their derogatory use [1, 2, 3, 4]. This linguistic phenomenon poses a significant challenge for automatic language processing systems [5]. Standard approaches for hate speech and abusive language detection often rely on offensive terms or expressions that are commonly used in hateful messages [6]. As a consequence, systems may incorrectly classify reclaimed language as harmful, leading to biased moderation outcomes and the silencing of marginalized voices. Addressing this issue requires models capable of capturing subtle semantic and contextual cues beyond the presence of potentially offensive terms [7, 8, 9]. The MultiPRIDE shared task [10] was proposed to foster research on reclaimed language within the LGBTQ+ community. The task formulates a binary classification problem aimed at determining whether a term related to the LGBTQ+ context in a message is used with a reclamatory intent or not. MultiPRIDE is articulated into two main tasks: Task A, which focuses on the textual content of the message, and Task B, which additionally allows the use of contextual information derived from the author's profile. Both tasks are further divided into language-specific subtasks covering Italian, Spanish, and English.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ greta.damo@univ-cotedazur.fr (G. Damo); n.b.ocampo@cw.nl (N. B. Ocampo)

🌐 <https://grexit-d.github.io/damo.greta.github.io/> (G. Damo); <https://www.nicolasbenjaminocampo.com/> (N. B. Ocampo)

🆔 0009-0009-8204-9513 (G. Damo); 0009-0001-0077-4626 (N. B. Ocampo)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

In this paper, we describe our participation in the MultiPRIDE shared task, addressing both Task A (textual content) and Task B (contextual content) for Italian and Spanish. We extend transformer-based models by incorporating two additional feature sets. First, we hypothesize that reclaimed slurs are frequently used with non-negative sentiment. To capture this aspect, we integrate an external sentiment analysis tool and include both the predicted sentiment label and its confidence score as additional model features. Second, we hypothesize that linguistic patterns such as the use of first-person pronouns and specific verb types may signal reclaiming intent. We therefore automatically detect these linguistic cues, calculate their frequency, and embed them as model features. Our results show consistent improvements over the baseline in Task A for both Italian and Spanish. In Task B, improvements are observed for Spanish, though not for Italian. We rank 8th out of 24 submissions for Italian and 6th out of 19 submissions for Spanish in Task A. In Task B, we place 11th out of 18 submissions for Italian and 5th out of 12 submissions for Spanish. These results suggest that the proposed features contribute positively to reclaimed slur detection and can enhance model performance beyond standard transformer-based approaches.¹

2. Methodology

In this section we describe the data used for the task, that were provided by the organizers, and how we splitted them into train, dev, test, and the systems we utilized.

2.1. Dataset

The datasets used in this work were provided by the organizers of the MultiPRIDE shared task. The multilingual corpus covers three languages—Italian, Spanish, and English—and was constructed by combining existing resources collected between 2020 and 2022 from social networks, blog posts, and TV series. For Italian, data were sourced from the TWITA collection [11], while Spanish data were drawn from a dataset of LGBTQ+-related messages [12]. The English portion aggregates content from multiple sources, including Twitter, Reddit, and TV series dialogues [13]. To build the final datasets, the organizers applied a consistent filtering methodology across languages. First, keyword-based filtering was performed using terms related to homosexuality extracted from the Hurltlex lexicon [14]. Subsequently, the selection was refined by identifying messages more likely to contain reappropriative language, based on the presence of positive terms associated with pride and community belonging (e.g., pride, queer, LGBT, rainbow). The final datasets consist of 1,811 instances for Italian, 1,461 for Spanish, and 1,711 for English. For our experiments, we focused on Italian and Spanish, in line with our participation in both Task A and Task B for these two languages. Starting from the training data released by the organizers, we created *train*, *development*, and *test* splits combining the instances of the 3 languages provided. We use a stratified sampling strategy to preserve label distribution. Specifically, we allocated 70% of the data for training, and split the remaining 30% evenly into development and test sets. All splits were generated using a fixed random seed to ensure reproducibility. Table 1 shows the label distribution of each split. Throughout the paper, we use the terms training, development, and test sets to refer to the data splits derived from the portion of the dataset released by the organizers during the “Data Release” phase of the MultiPRIDE task. We refer to the split retained by the organizers and used during the “Evaluation Window” as the *submission* set. It is important to note that this set is not accessible to participants, and that only two systems per task and per language could be submitted for evaluation. Moreover, although we officially participated in the Italian and Spanish tracks, we also included English-language instances in our data splits in order to leverage them during training. This decision was motivated by the very limited number of instances containing reclaimed slurs: by combining data from all three languages, we obtained 300 such instances to be used as training material. However, we decided to participate for only Spanish and Italian.

¹The code, trained models, and instructions for reproducing our experiments are publicly available at: https://github.com/grexit-d/HateItOff_MultiPRIDE.

Lang	Label	Train		Dev		Test		Submission
		#	%	#	%	#	%	
EN	recl	62	0,0864	13	0,0844	13	0,0844	685
	non-recl	656	0,9136	141	0,9156	141	0,9156	
ES	recl	93	0,1517	20	0,1527	20	0,1515	585
	non-recl	520	0,8483	111	0,8473	112	0,8485	
IT	recl	145	0,1908	31	0,1902	31	0,1902	725
	non-recl	615	0,8092	132	0,8098	132	0,8098	

Table 1

Data splits and label distribution for text messages labeled as reclaimed (recl) and non reclaimed (non-recl) of the MultiPRIDE Shared Task dataset.

2.2. Description of the System

For our participation in the MultiPRIDE shared task, we submitted two systems per task and per language. We adopted an ablation-based approach to identify the most suitable systems for submission. Our ablation study starts from a vanilla transformer-based model and incrementally incorporates additional features. These features fall into two main categories: linguistic features and sentiment features. Each resulting system configuration is described below.

Baseline. All the proposed systems rely on pre-trained transformer encoders fine-tuned for binary sequence classification. Given an input message, the model predicts whether a term related to the LGBTQ+ context is used with a reclamatory intent.

Sentiment-enriched models. We hypothesize that the use of reclaimed slurs tends to rely on non-negative sentiment. In this direction, we trained transformer models with injected sentiment label and sentiment score from the sentiment analysis model `tabularisai/multilingual-sentiment-analysis`. The predicted sentiment label and/or the sentiment score are appended to the original input text in the form of a natural-language prompt. This enables the classifier to use sentiment cues without modifying the transformer encoder architecture.

Linguistically-enriched models. We hypothesize that reclamation is often associated with the use of first-person personal pronouns, such as “yo/nosotros” in Spanish and “io/noi” in Italian, as well as with specific call-to-action expressions, for example “*Speak up, queer voices matter.*”. Therefore, we experimented with linguistic features automatically extracted using `SPaCy`, including counts and normalized ratios of personal pronouns (first, second, and third person), the number of verbs, and the presence of imperative forms. The extracted linguistic features are concatenated with the transformer’s [CLS] representation and fed into a feed-forward classification head.

Implementation details and ablation study. We used BERT, ModernBERT, and XLM as our baseline models. These models serve as backbones, later enriched with the three strategies described: sentiment label, sentiment score, and linguistic cues. We selected these models to maintain simple, adaptable strategies that could later incorporate additional features. To measure the impact of each strategy while minimizing noise, we preferred these transformer models over LLMs. The ablation study considers the set of strategies $S = \{\text{sentiment label, sentiment score, linguistic cues}\}$ and all possible subsets of S , i.e., the power set $\mathcal{P}(S)$. Using the set of backbone models $M = \{\text{BERT, ModernBERT, XLM}\}$, we compute the Cartesian product of M and $\mathcal{P}(S)$. Considering two tasks, A and B, this results in $3 \text{ models} \times 8 \text{ strategy subsets} \times 2 \text{ tasks} = 48 \text{ runs}$.

All models were implemented using the Hugging Face Transformers library. For BERT, ModernBERT, and XLM models, we used the `bert-base-uncased`, `answerdotai/ModernBERT-base`, and `cardiffnlp/twitter-xlm-roberta-base` versions available on Hugging Face, respectively. Training and evaluation were performed with a batch size of 8. Models were evaluated and checkpointed at the end of each epoch, retaining up to two checkpoints, and the best-performing model was selected based on the macro-averaged F1-score on the development set. Training logs were recorded every 50 steps. All experiments were conducted with a fixed random seed on a single NVIDIA A100 GPU. Experiments

Task	Submitted Systems	Spanish			Italian		
		no-recl	recl	macro	no-recl	recl	macro
A ES	S1 BERT + ling. cues	0,9335	0,5405	0,7370	-	-	-
	S2 ModernBERT + sent. label. + sent. scores	0,9234	0,3651	0,6442	-	-	-
A IT	S1 BERT + ling. cues	-	-	-	0,9571	0,8046	0,8809
	S2 UmBERTo + ling. cues	-	-	-	0,9549	0,7619	0,8584
B ES	S1 BERT + ling. cues + sent. label	0,9247	0,4762	0,7005	-	-	-
	S2 XLMT + sent. label + sent. scores	0,9280	0,4186	0,6733	-	-	-
B IT	S1 ModernBERT + sent. label	-	-	-	0,9416	0,7222	0,8319
	S2 ModernBERT + sent. label. + sent. scores	-	-	-	0,9497	0,7426	0,8462

Table 2

Results of System 1 and 2 on tasks A and B for Spanish and Italian on the submission test set.

were run for both 3-epoch and 6-epoch training schedules, resulting in $48 \times 2 = 96$ runs. In addition to the ablation study, we also evaluated for Task A four additional configurations using UmBERTo, an extension of the RoBERTa model for Italian trained with Whole Word Masking. The evaluated configurations include UmBERTo with sentiment label only, linguistic cues only, and the combination of sentiment label and linguistic cues. The motivation behind these experiments was to assess whether an Italian-specific language model could further improve performance on the task. In total, we conducted 100 runs for all the systems proposed in this shared task.

3. Results

The results of the ablation study across all experimental runs are reported in Table 3 for the 3-epoch setting and Table 4 for the 6-epoch setting (both tables are provided in Appendix A). The results indicate that certain features, such as linguistic cues and sentiment labels, have a positive impact on the prediction of slur reclamation, particularly for BERT and ModernBERT models. Regarding the differences between Task A and Task B, performance for Spanish decreases when contextual information is included, whereas for Italian it remains unchanged, yielding comparable results with and without context. In our experiments, incorporating users’ *BIO* information did not contribute to improved detection performance.

We selected models for submission based on both performance over the baseline and system diversity. Although we could have chosen the two best-performing configurations for each task and language, we aimed to reduce the risk of selecting models that performed well on our internal test split but might not generalize to the submission set.

Table 2 shows the selected systems with their results on the *submission* set. We can see that models perform particularly well on reclaim slur detection for Italian, achieving a macro score of 0.8809, with the S1 BERT + linguistic cues system being the best. For Spanish, the best system achieves a macro score of 0.7370. We also observe that our models generally perform better when no context is used as seen in the ablation study.

4. Error Analysis

4.1. Italian Language

Manual inspection of the misclassified Italian messages reveals that errors can be grouped into a limited number of recurring categories. These categories highlight both annotation ambiguities and modeling limitations when dealing with reclaimed language.

Borderline or Ambiguous Reclaimed Usage. Some messages contain slur-related terms in contexts that are neither clearly positive nor explicitly negative. In these cases, reclamation is ambiguous and highly dependent on pragmatic interpretation, making both annotation and automatic classification difficult. Examples:

1. *"sono sempre loro che ci adescano mannaggia a loro, con le loro frociate..."* (it_887);
2. *"Che carini tutti questi che sono spuntati adesso a difendere i diritti dei froci."* (it_502).

These messages adopt an ironic or critical tone, and the reclaimed intent is not explicit, suggesting potential inconsistencies in the gold labels.

Reclaimed Slurs Combined with Negative or Aggressive Language Several instances show reclaimed self-references co-occurring with insults, profanity, or aggressive discourse directed at others. While the slur may be reclaimed, the overall negative tone can mislead sentiment- or toxicity-aware models. Examples:

3. *"Orgoglioso di essere 'frocio'! Omofoba di merda!"* (it_1750);
4. *"Ripetiamolo insieme non potete dire frocio se non siete lgbtq+... Are u that stupid?"* (it_474);
5. *"Ci sono 'finocchi' che lo fanno orgogliosamente col culo proprio."* (it_695).

These cases highlight the difficulty of disentangling reclaimed identity markers from general offensive language.

In-Group Criticism and Community-Internal Discourse Some misclassified messages contain reclaimed slurs used within critical reflections on the LGBTQ+ community itself. Although reclamation is present, the critical stance complicates interpretation. Examples:

6. *"Omofobia interiorizzata da parte di froci che si credono etero..."* (it_1057);
7. *"alle frocie queer non le fanno né partecipare..."* (it_1242).

Here, reclaimed terms are used for internal critique rather than affirmation, a phenomenon that current models struggle to capture.

Quotation, Meta-Discussion, and Explanatory Uses In these cases, slurs appear inside quotations, explanations, or meta-linguistic discussions about their meaning or usage, rather than being directly employed as insults or self-labels. Examples:

8. *"il termine frocio sia attribuito a qualcuno negativamente..."* (it_1179);
9. *"'Frocio' me l'hanno fatto uscire dalle orecchie quando andavo a scuola"* (it_1401);
10. *"Nuovo epiteto ricevuto degno di nota..."* (it_979).

These uses are typically reclaimed or reflective, but surface-level lexical cues can trigger misclassification.

Pragmatic and Contextual Cues Beyond Textual Content Some messages rely heavily on emojis, hashtags, or shared cultural context (e.g., Pride events), which are not always fully exploited by text-based models. Examples:

11. *"Buongiorno, specialmente ai miei amici froci..."* 🏳️‍🌈 🏳️‍🌈 ❤️ (it_631);
12. *"🏳️‍🌈 🏳️‍🌈 Fatelo per lui..."* (it_1401).

Although clearly supportive or reclaimed to a human reader, these signals may be underweighted by the model.

Community Membership Without Explicit Reclaimed Slurs Finally, some false positives occur in messages where the author self-identifies as part of the LGBTQ+ community, but no reclaimable slur is actually used. Examples:

13. *"un'attrice lella che seguo è incinta..."* (it_1222);
14. *"Le persone trans sono a volte guidate dall'omofobia..."* (it_269).

These cases suggest that models may over-rely on topical cues or community-related vocabulary.

Overall, the error analysis shows that misclassifications are often due to pragmatic ambiguity, mixed sentiment, and meta-linguistic usage of slurs. These findings confirm that reclaimed language detection cannot be reduced to lexical or sentiment-based signals alone and requires deeper modeling of speaker intent, discourse function, and community context.

4.2. Spanish Language

We also inspected the misclassified Spanish messages. They can be grouped into the following recurring categories:

Criticism and Non-Textual Content In Spanish, messages containing reclaimed slurs often rely on emojis and hashtags commonly used to label LGBTQ+ topics or to signal in-group membership. However, these same markers may also appear in messages that criticize or oppose the community, even from individuals who explicitly claim membership in it. Examples:

15. “No se quien sea el organizador del PRIDE 🏳️ 2022 pero en que cabeza cabe invitar a #RominaMarcos que ella y su familia son homofóbicos misoginos y machistas no conforme con eso invitan al señor #Alfredoadame que usa palabras como joto y marica para ofender #LGTBI”. (es_535).
16. “”@USER Esto me da a mí que roza la ilegalidad, uds sois una entidad oficial de un estado con bandera 🇪🇸, no sois circo mariconchi, vergüenza. Marlaska no puede dar más asco. #LGTBI #PRIDE2020 PD. soy gay.” (es_1872).

Exaggeration Some other instances use intensifiers or exaggeration to deliver hate to the community.

17. “**Primero un dia y ahora un mes.** Al final el dia de los maricas sera dia no laborable #OrgulloLGTBI”. (es_465).

Sarcasm There are slurs that might look as if they were used by members of the community but actually were used with sarcasm.

18. “Feliz día de la mariconeria #LGTBI #Pride2022” (es_372).
19. “**YO SOY EL COLECTIVO!!!!** #OrgulloLGTBI **pues si**, hasta que llegasteis las maricas **rojas-podemitas-fascistoides**, la GS era una reivindicación del colectivo, y a punto estuvimos de conseguirlo como los israelíes. (es_275). F-I-N URL”

“mariconeria” is a slight variation of the reclaimed slur “maricón” that is used with negative connotations while “YO SOY EL COLECTIVO” is a way of mocking which is later revealed with by calling the “rojas-podemitas-fascistoides”

5. Conclusion

In this study, we focus on the detection of messages containing reclaimed LGBTQ+ slurs within the MultiPRIDE shared task. We participated in Task A, which relies solely on the single instance of the message, and Task B, which incorporates both the message and its context to assess whether contextual information improves detection. Our team, *HateItOff*, participated in both tasks for Spanish and Italian. Our proposal and underlying hypothesis is that reclamation typically depends on a speaker who belongs to the targeted community using a reclaimed slur to express belongingness or identity. A common way this is expressed in natural language is through the use of pronouns and verbs, usually in a non-negative manner. Based on this intuition, we experimented with transformer models that incorporate linguistic features and sentiment features, providing cues related to pronoun and verb usage as well as the overall sentiment of the message. We conducted an Ablation Study and observed that adding these features resulted in a performance boost over the baseline for both languages. For Task A, our best-performing system surpassed the baselines provided by the organizers, ranking 8th out of 24 submissions for Italian and 6th out of 19 submissions for Spanish. For Task B, we achieved 11th place out of 18 submissions for Italian and 5th out of 12 submissions for Spanish. In this task, our system outperformed the organizers’ baseline for Spanish only.

Acknowledgments

This work has been supported by the French government, through the 3IA Cote d’Azur investments in the project managed by the National Research Agency (ANR) with the reference number ANR-23-IACL-0001. This work was also carried out during the tenure of an ERCIM ‘Alain Bensoussan’ Fellowship Programme.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] D. L. de Sa, B. Cepollaro, The successes of reclamation, *Synthese* 202 (2023) 1–19. URL: <https://philpapers.org/rec/DESTSO-18>. doi:10.1007/s11229-023-04393-y.
- [2] C. Bianchi, Slurs and appropriation: An echoic account, *Journal of Pragmatics* 66 (2014) 35–44. URL: <https://www.univr.it/attachments/Bianchi-C.-2014-9C%20Slurs-and-appropriation--an-echoic-account%20%9D,-Journal-of-Pragmatics-66,-pp.-35-44,-DOI-10.1016-j.pragma.2014.02.009./be0c1563-7e7d-47b4-87ee-996c68fcf241/70483ac4-c1d0-4514-ae7b-be89664df395.pdf>. doi:10.1016/j.pragma.2014.02.009, published in *Journal of Pragmatics* 66:35–44. Copyright Elsevier B.V.
- [3] E. Nossem, Queer, frocia, femminiellə, ricchione et al. – localizing “queer” in the italian context, *gender/sexuality/italy* 6 (2019). URL: <https://www.gendersexualityitaly.com/1-queer-frocia-femminiell%C9%99-ricchione-et-al-localizing-queer-in-the-italian-context/>. doi:10.15781/31yc-ys20.
- [4] B. Cepollaro, Linguaggio d’odio, in: *Linguaggio d’odio*, pp. 145–156, 2022. URL: <https://hdl.handle.net/20.500.11768/144456>.
- [5] D. O. Thiago, A. D. Marcelo, A. Gomes, Fighting hate speech, silencing drag queens? artificial intelligence in content moderation and risks to lgbtq voices online, *Sexuality & Culture* 25 (2021) 700–732.
- [6] N. B. Ocampo, E. Sviridova, E. Cabrio, S. Villata, An in-depth analysis of implicit and subtle hate speech messages, in: A. Vlachos, I. Augenstein (Eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 1997–2013. URL: <https://aclanthology.org/2023.eacl-main.147/>. doi:10.18653/v1/2023.eacl-main.147.
- [7] E. W. Pamungkas, V. Basile, V. Patti, Do you really want to hurt me? predicting abusive swearing in social media, in: *Proceedings of the Twelfth Language Resources and Evaluation Conference (LREC 2020)*, European Language Resources Association, Marseille, France, 2020, pp. 6237–6246. URL: <https://aclanthology.org/2020.lrec-1.765>.
- [8] E. W. Pamungkas, V. Basile, V. Patti, Investigating the role of swear words in abusive language detection tasks, *Language Resources and Evaluation* 57 (2023) 155–188. URL: <https://doi.org/10.1007/s10579-022-09582-8>. doi:10.1007/s10579-022-09582-8.
- [9] L. Draetta, C. Ferrando, M. Cuccarini, L. James, V. Patti, Reclaim project: Exploring italian slurs reappropriation with large language models, in: *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, 2024, pp. 335–342.
- [10] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [11] V. Basile, M. Lai, M. Sanguinetti, et al., Long-term social media data collection at the university of

- turin, in: Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR-WS, 2018, pp. 1–6.
- [12] J. Mata, E. Gualda, A dataset of spanish tweets on people and communities lgbtqi+ during the covid-19 pandemic 2020-2022 [lgbtqi+ dataset 2020-2022_es], 2025. Dataset.
- [13] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: Proceedings of the 16th ACM Web Science Conference, ACM, 2024, pp. 241–249.
- [14] E. Bassignana, V. Basile, V. Patti, et al., Hurtlex: A multilingual lexicon of words to hurt, in: CEUR Workshop Proceedings, volume 2253, CEUR-WS, 2018, pp. 1–6.

A. Ablation Study for System Selection

In this section we show the results of the ablation study carried out between the transformer models used and the feature strategies. Table 3 and 4 show the results on the test set that we used for our experiments with transformers trained on 3 and 6 epochs. Table 5 shows our additional experiments carried out with UmBERTo for Task A.

3 Epochs Training with Data in Spanish, Italian, and English							
Task	Model	Spanish			Italian		
		no-recl	recl	macro	no-recl	recl	macro
Task A	bert_multil	0,9231	0,4000	0,6615	0,9627	0,8276	0,8951
	+ ling. cues	0,9224	0,4375	0,6800	0,9734	0,8889	0,9311
	+ sent. score	0,9099	0,3226	0,6162	0,9434	0,7541	0,8487
	+ sent. label	0,9231	0,4000	0,6615	0,9704	0,8571	0,9138
	+ sent. score	0,9145	0,3333	0,6239	0,9368	0,7018	0,8193
	+ sent. label	0,9091	0,3636	0,6364	0,9635	0,8077	0,8856
	+ sent. score	0,9099	0,3226	0,6162	0,9697	0,8710	0,9203
	+ sent. score	0,9185	0,3871	0,6528	0,9742	0,8727	0,9234
	modernbert_multil	0,8987	0,3784	0,6385	0,9509	0,7869	0,8689
	+ ling. cues	0,9153	0,2857	0,6005	0,9552	0,7931	0,8742
	+ sent. score	0,9185	0,3871	0,6528	0,9670	0,8302	0,8986
	+ sent. label	0,8996	0,3429	0,6212	0,9630	0,8214	0,8922
	+ sent. score	0,9185	0,3871	0,6528	0,9632	0,8148	0,8890
	+ sent. label	0,9099	0,3226	0,6162	0,9627	0,8276	0,8951
	+ sent. score.	0,9217	0,4706	0,6962	0,9663	0,8475	0,9069
	+ sent. score	0,9130	0,4118	0,6624	0,9478	0,7586	0,8532
	xlmt_multil	0,9177	0,4242	0,6710	0,9699	0,8667	0,9183
	+ ling. cues	0,9198	0,2963	0,6081	0,9429	0,6522	0,7975
	+ sent. score	0,9198	0,2963	0,6081	0,9734	0,8889	0,9311
	+ sent. label	0,9018	0,4500	0,6759	0,9655	0,8615	0,9135
	+ sent. score	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. label	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score.	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score	0,9205	0,2400	0,5803	0,9736	0,8852	0,9294
Task B	bert_multil	0,9004	0,3030	0,6017	0,9704	0,8571	0,9138
	+ ling. cues	0,9043	0,3529	0,6286	0,9701	0,8621	0,9161
	+ sent. score	0,9277	0,4138	0,6707	0,9738	0,8814	0,9276
	+ sent. label	0,9185	0,3871	0,6528	0,9699	0,8667	0,9183
	+ sent. score	0,9362	0,4828	0,7095	0,9478	0,7586	0,8532
	+ sent. label	0,9205	0,2400	0,5803	0,9630	0,8214	0,8922
	+ sent. score	0,9205	0,2400	0,5803	0,9738	0,8814	0,9276
	+ sent. score	0,9198	0,2963	0,6081	0,9627	0,8276	0,8951
	modernbert_multil	0,9130	0,4118	0,6624	0,9776	0,8966	0,9371
	+ ling. cues	0,9145	0,3333	0,6239	0,9738	0,8814	0,9276
	+ sent. score	0,9237	0,3571	0,6404	0,9776	0,8966	0,9371
	+ sent. label	0,9060	0,2667	0,5863	0,9742	0,8727	0,9234
	+ sent. score	0,9198	0,2963	0,6081	0,9888	0,9492	0,9690
	+ sent. label	0,8987	0,3784	0,6385	0,9630	0,8214	0,8922
	+ sent. score	0,8938	0,3684	0,6311	0,9848	0,9355	0,9602
	+ sent. score	0,8996	0,3429	0,6212	0,9697	0,8710	0,9203
	xlmt_multil	0,9145	0,3333	0,6239	0,9701	0,8621	0,9161
	+ ling. cues	0,9129	0,0870	0,4999	0,9701	0,8621	0,9161
	+ sent. score	0,9167	0,1667	0,5417	0,9263	0,4878	0,7071
	+ sent. label	0,9145	0,3333	0,6239	0,9588	0,8136	0,8862
	+ sent. score	0,9198	0,2963	0,6081	0,9776	0,8966	0,9371
	+ sent. label	0,9205	0,2400	0,5803	0,9559	0,7778	0,8668
	+ sent. score	0,9231	0,4000	0,6615	0,9520	0,7636	0,8578
	+ sent. score	0,9121	0,1600	0,5361	0,9559	0,7778	0,8668

Table 3
Ablation Study on Tasks A and B for the Spanish and Italian languages using models trained with 3 epochs.

6 Epochs Training with Data in Spanish, Italian, and English							
Task	Model	Spanish			Italian		
		no-recl	recl	macro	no-recl	recl	macro
Task A	BERT	0,9244	0,5641	0,7443	0,9430	0,7619	0,8524
	+ ling. cues	0,9264	0,4848	0,7056	0,9582	0,8254	0,8918
	+ sent. score	0,8957	0,2941	0,5949	0,9627	0,8276	0,8951
	+ sent. label	0,9292	0,5789	0,7541	0,9621	0,8387	0,9004
	+ sent. score	0,9130	0,4118	0,6624	0,9509	0,7869	0,8689
	+ sent. label	0,9091	0,3636	0,6364	0,9585	0,8197	0,8891
	+ sent. score	0,9310	0,5000	0,7155	0,9398	0,7333	0,8366
	+ sent. score	0,9153	0,2857	0,6005	0,9774	0,9000	0,9387
	ModernBERT	0,8966	0,2500	0,5733	0,9489	0,7308	0,8398
	+ ling. cues	0,9099	0,3226	0,6162	0,9430	0,7619	0,8524
	+ sent. score	0,8978	0,4103	0,6540	0,9552	0,7931	0,8742
	+ sent. label	0,9170	0,4571	0,6871	0,9506	0,7937	0,8721
	+ sent. score	0,9185	0,3871	0,6528	0,9738	0,8814	0,9276
	+ sent. label	0,9030	0,1481	0,5256	0,9630	0,8214	0,8922
	+ sent. score	0,9123	0,4444	0,6784	0,9627	0,8276	0,8951
	+ sent. score	0,9224	0,4375	0,6800	0,9742	0,8727	0,9234
	XLM	0,9224	0,4375	0,6800	0,9697	0,8710	0,9203
	+ ling. cues	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. label	0,9185	0,3871	0,6528	0,9615	0,8485	0,9050
	+ sent. score	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. label	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score	0,9115	0,4737	0,6926	0,9494	0,8116	0,8805
	+ sent. score	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
Task B	BERT	0,9270	0,4516	0,6893	0,9588	0,8136	0,8862
	+ ling. cues	0,8918	0,2424	0,5671	0,9660	0,8525	0,9092
	+ sent. score	0,9304	0,5294	0,7299	0,9549	0,8000	0,8774
	+ sent. label	0,9217	0,4706	0,6962	0,9736	0,8852	0,9294
	+ sent. score	0,9106	0,2759	0,5933	0,9704	0,8571	0,9138
	+ sent. label	0,9174	0,0909	0,5041	0,9531	0,7347	0,8439
	+ sent. score	0,9205	0,2400	0,5803	0,9565	0,7600	0,8583
	+ sent. score	0,9283	0,3704	0,6493	0,9813	0,9153	0,9483
	ModernBERT	0,9091	0,3636	0,6364	0,9701	0,8621	0,9161
	+ ling. cues	0,9091	0,3636	0,6364	0,9699	0,8667	0,9183
	+ sent. score	0,9138	0,3750	0,6444	0,9740	0,8772	0,9256
	+ sent. label	0,9099	0,3226	0,6162	0,9740	0,8772	0,9256
	+ sent. score	0,9138	0,3750	0,6444	0,9738	0,8814	0,9276
	+ sent. label	0,9091	0,3636	0,6364	0,9850	0,9333	0,9591
	+ sent. score	0,8860	0,2778	0,5819	0,9701	0,8621	0,9161
	+ sent. score	0,8850	0,3158	0,6004	0,9773	0,9032	0,9402
	XLM	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ ling. cues	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score	0,9114	0,2222	0,5668	0,9630	0,8214	0,8922
	+ sent. label	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score	0,9160	0,2308	0,5734	0,9774	0,9000	0,9387
	+ sent. label	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475
	+ sent. score	0,9060	0,2667	0,5863	0,9740	0,8772	0,9256
	+ sent. score	0,9180	0,0000	0,4590	0,8949	0,0000	0,4475

Table 4

Ablation Study on Tasks A and B for the Spanish and Italian languages using models trained with 6 epochs.

Model	no-recl	recl	macro
UmBERTo	0,9542	0,8125	0,8833
UmBERTo + sent. label	0,9582	0,8254	0,8918
UmBERTo + ling. cues	0,9704	0,8571	0,9138
UmBERTo + ling. cues + sent. label	0,9663	0,8475	0,9069

Table 5
Additional UmBERTo runs on Task A

I2C At MultiPRIDE: Transformers And Generative LLMs For Multilingual Reclamation Classification

Laura Vázquez-Ramos^{1,*†}, Jacinto Mata-Vázquez^{1,†} and Victoria Pachón-Álvarez^{1,†}

¹I2C Research Group, Information Technology Department, Universidad de Huelva, Fuerzas Armadas Avenue, 21007, Huelva, Spain

Abstract

Online hate and abusive language targeting LGBTQ+ people are increasingly common on social media. Such content can spread prejudice, intimidate users, and limit participation in online spaces. Automatic detection is therefore important to support moderation and to reduce exposure to harmful content. This paper presents our submissions to MultiPRIDE 2026 Task 1 for Italian, Spanish, and English, comparing two modelling paradigms: an encoder-based Transformer sequence classifier and a generative large language model adapted to the task through supervised instruction tuning. Results are reported with macro-averaged F1 to account for class imbalance. Considering the best-performing run per subtask, our system achieves macro-F1 = 0.571 (EN), 0.711 (ES), and 0.843 (IT) on the official test sets, showing the strongest performance in Italian and consistent improvements across languages. These results suggest that discriminative multilingual Transformers remain a robust choice for intent-sensitive reclamation detection under limited context.

Content Warning: This paper contains examples of explicit and offensive language, including hate speech and slurs, presented for scientific analysis and documentation purposes.

Keywords

Natural Language Processing, LGBTQ+ Hate Speech, Abusive Language Detection, Transformers, Large Language Models

1. Motivation

Online texts often include terms related to LGBTQ+ identity. Sometimes they are used as insults. Other times they are used by the community itself, as self-labels or jokes. This second case is known as reclaimed language: words that were once offensive but are re-used with a positive or in-group meaning [1].

Distinguishing between reclamatory and offensive usage is important for both moderation and discourse analysis. If reclaimed language is not recognised, legitimate messages may be flagged as harmful, resulting in the disproportionate removal of LGBTQ+ voices. Conversely, if hostile uses are misinterpreted as reclamatory, targeted abuse may remain undetected. For this reason, the central challenge is not simply the presence of specific terms, but the intent with which they are used.

This problem is challenging because intent is highly context-dependent. It may be conveyed indirectly through irony, euphemisms, or figurative language, and its interpretation can vary across languages and communities. Under text-only conditions—without conversational history or user-level information—key pragmatic cues are missing, increasing ambiguity and making reliable classification difficult.

Within this context, the MultiPRIDE 2026 Task 1 [2] provides a multilingual benchmark for evaluating systems that decide whether an LGBTQ+ related term in a sentence is used with reclamatory intent or not, across Italian, Spanish, and English. The goal of this work is to contribute to this evaluation by developing and analysing two complementary modelling paradigms—Transformer-based sequence classification and generative LLM-based inference—and by examining the factors that most strongly affect performance across languages.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ laura.vazquez@dti.uhu.es (L. Vázquez-Ramos); mata@uhu.es (J. Mata-Vázquez); vpachon@uhu.es (V. Pachón-Álvarez)

ORCID 0009-0001-3974-8943 (L. Vázquez-Ramos); 0000-0001-5329-9622 (J. Mata-Vázquez); 0000-0003-0697-4044

(V. Pachón-Álvarez)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

id	text	label
it_1231	La destra Italiana pur di non dire che loro odiano gli omosessua...rola Identità di Genere.....ovviamente a la #Salvini #omnibusla7	0
it_1713	"Presupporre che tutti i bisessuali non sono mai attratti da per... e genderqueer che scelgono di etichettarsi come bisessuali" URL	0
it_1770	io devo sapere chi è l'intern forcio che si occupa della musica a cortesie per gli ospiti URL	1
it_1155	A volte mi ricordo di quanto sia bello essere lella e non poterlo dire con fierezza a nessuno	1

Table 1

Examples from the Italian dataset (Subtask A1).

2. Definition of the Task

This work addresses MultiPRIDE 2026 Task 1, which focuses on reclaimed language in the LGBTQ+ community. The goal is to study how terms related to LGBTQ+ identity are used in online text and to highlight the challenges that arise when a term can be either derogatory or used with a reclamatory, in-group intent. Systems are expected to rely on cues that can be expressed in the text itself (e.g., slurs, denigratory words, self-labeling, figurative language), and, in general, to capture the intended meaning behind the use of these terms.

Task 1 is formulated as a binary classification problem. Given a sentence that contains a term related to an LGBTQ+ context, the system must predict whether the term is used with reclamatory intent or non-reclamatory intent.

We participated in the text-only setting (Task A – Textual Content), where the input consists solely of the message text and no user-profile information is provided. Participation was carried out in all three language-specific subtasks:

- **Subtask A1 (Italian):** classification on Italian texts.
- **Subtask A2 (Spanish):** classification on Spanish texts.
- **Subtask A3 (English):** classification on English texts.

3. Dataset

The organisers released one dataset per language track. Each instance includes an identifier (*id*), the message text (*text*), and a binary label (*label*) indicating whether the LGBTQ+-related term is used with reclamatory intent (1) or non-reclamatory intent (0). A language tag (*lang*) is also provided. For Italian and Spanish, an additional field (*bio*) is included; however, it was not used in our experiments because we participated in the text-only setting.

Three monolingual datasets were provided, corresponding to Italian (Subtask A1), Spanish (Subtask A2), and English (Subtask A3). The Italian dataset contains 1,086 instances, with 879 (80.94%) labelled as class 0 and 207 (19.06%) as class 1. The Spanish dataset contains 876 instances, with 743 (84.82%) instances in class 0 and 133 (15.18%) in class 1. The English dataset contains 1,026 instances and is the most imbalanced split, with 938 (91.42%) instances labelled as class 0 and 88 (8.58%) as class 1.

In addition to the three monolingual datasets, a multilingual training set was created by merging Italian, Spanish, and English into a single pool to support multilingual training. This setup was treated as an unconstrained approach, since an additional derived dataset was constructed for training purposes. Before balancing, the merged dataset contained 2,988 instances in total, with 2,560 instances in class 0 and 428 instances in class 1.

Illustrative examples for each language are reported in Tables 1, 2, and 3, respectively. Table 4 shows the class distribution of the three monolingual datasets and the merged multilingual dataset, reporting the number of instances for class 0 and class 1 in each case.

id	text	label
es_1850	28 de Junio - Día Internacional del Orgullo LGTBI. Viva el amor Libre!!!#lgtb #lgtbiq #lgtbi #amorlibre #loveislove #orgullogay #pride #...#lovewins #orgullo #respeto #lucha #homosexual #doblaespain URL	0
es_773	@USER no me gusta la Montero, por su apoyo a la ley Trans, ignor... es un acoso y derribo a una mujer que lucha por sus hijos.. URL	0
es_1196	¡¡¡Ya estamos de MARlcañas y BOLLOpintxos!!!#Orgullo2021 #Orgullo-Rioja2021 #OrgulloLGTBI URL	1
es_1884	Soy Queer.#Queer #Orgullo #Pride #Orgullo2020 #OrgulloLGTBI #LGT-BIQ #trescantos URL	1

Table 2

Examples from the Spanish dataset (Subtask A2).

id	text	label
en_1021	I've never heard anyone use the word "faggot" and not sound like an insecure 13 year old.	0
en_1496	So you don't see the slightest problem of somebody calling other people "faggots"?	0
en_1312	And to be fair, getting triggered by slurs is just stupid to begin with. Yeah I'm a faggot, yeah I'm gay. I'm mo...I because it has more syllables and thus takes more time to say.	1
en_1108	Tbh I don't care if anyone says it as long as they say it jokingly. Also happy I'm not the only one who calls themselves faggot	1

Table 3

Examples from the English dataset (Subtask A3).

Dataset	Class 0	Class 1	Total
Italian (A1)	879	207	1086
Spanish (A2)	743	133	876
English (A3)	938	88	1026
Multilingual (IT+ES+EN)	2560	428	2988

Table 4

Class distribution for the three monolingual datasets and the merged multilingual dataset (before balancing).

4. Description of the System

This section describes the methodology adopted, which is structured into two main stages: balancing and splitting the datasets and training models.

4.1. Balancing and Splitting the Datasets

For development, an internal 80/20 split was applied to obtain training and evaluation partitions. Since the original datasets are strongly imbalanced, a data augmentation technique based on Contextual Word Embeddings (CWE) was applied only on the training split [3]. Specifically, we utilized a BERT-based architecture, the bert-base-uncased model [4]. This method entails substituting selected tokens with semantically similar alternatives derived from the model's contextual representations, thereby increasing the lexical diversity of the corpus while preserving the semantic integrity of the original instances. In this case, part of the majority class (label 0) was removed and new examples of the minority class (label 1) were generated until reaching 1,500 instances per label. As a result, the balanced training sets contain 3,000 instances each.

Before applying CWE-based balancing, each dataset was split using an 80/20 scheme. The 80% portion was used for model development, while the remaining 20% was kept as an unseen test set for the final evaluation. Table 5 reports the class distribution for each subtask before and after applying

Dataset	Label	Before balancing (Train)	After balancing (Train)
A1 (Italian)	0 (non-reclamatory)	703	1500
	1 (reclamatory)	165	1500
A2 (Spanish)	0 (non-reclamatory)	594	1500
	1 (reclamatory)	106	1500
A3 (English)	0 (non-reclamatory)	750	1500
	1 (reclamatory)	70	1500
Multilingual (IT+ES+EN)	0 (non-reclamatory)	2048	1500
	1 (reclamatory)	342	1500

Table 5

Label distribution in the training split before and after CWE-based balancing (balancing applied only to Train; the Test split was kept unchanged).

Dataset	Train	Valid	Test
A1 (Italian)	2400	600	218
A2 (Spanish)	2400	600	176
A3 (English)	2400	600	206
Multilingual (IT+ES+EN)	2400	600	598

Table 6

Number of instances in the training, validation, and test splits. CWE-based balancing was applied only to the initial Train split (80%).

data augmentation on the development portion only. The development split was then further divided using another 80/20 split to obtain the final training and validation sets. The validation set was used for model selection and hyperparameter tuning, whereas the test set was reserved exclusively for reporting results. Table 6 reports the number of instances assigned to the training and test partitions for each subtask.

4.2. Training models

Since the shared task allowed two runs per subtask, two complementary modelling approaches were developed for each language (Italian, Spanish, and English). A first run was based on a Large Language Model (LLM) [5] adapted to the task through supervised instruction tuning (QLoRA), where the model was guided to predict whether the target term was used with reclamatory intent or not. A second run relied on a Transformer model [6] using the provided training data, aiming to learn task-specific decision boundaries from labelled examples. This two-run setup was designed to compare a standard fine-tuning approach against a prompt-based alternative under the same evaluation protocol, and to assess how each paradigm behaves better in each subtask.

4.2.1. Run 1: QLoRA Fine-tuning

For the first run submitted for each subtask, an LLM-based approach was adopted. Models were fine-tuned with QLoRA [7], which enables parameter-efficient training by combining low-rank adapters with quantised weights, making it feasible to adapt LLMs under limited computational resources while preserving most of the base model’s knowledge. The main backbone used across experiments was *Qwen2.5-0.5B* [8], selected as a lightweight instruction-capable model that can be fine-tuned efficiently and deployed consistently across languages. In addition, a different backbone was used for Spanish: *teknum/OpenHermes-2.5-Mistral-7B*¹. This choice was motivated by the strong availability of Spanish-friendly instruction tuning in the Mistral/OpenHermes family and by empirical development results, where Spanish performance benefited from a larger, instruction-focused model. In practice, this design

¹<https://huggingface.co/teknum/OpenHermes-2.5-Mistral-7B>

Hyperparameter	Search space	Best value
Learning rate	{3e-5, 5e-5, 8e-5, 1e-4}	1e-4
Train batch size	{2, 4, 8}	2
Eval batch size	{16}	16
Gradient accumulation steps	{1, 2, 4}	4
LR scheduler	{linear, cosine}	linear
Warmup ratio	{0.03, 0.05, 0.1}	0.03
Weight decay	{0.0, 0.01}	0.01
Label smoothing	{0.0, 0.02}	0.0

Table 7

LLM hyperparameter search space and best configuration selected on the validation split.

provided a balance between efficiency and coverage (Qwen2.5-0.5B) and capacity for Spanish-specific instruction following (OpenHermes-2.5-Mistral-7B) [9].

A hyperparameter study was carried out for the LLM-based classifier in order to obtain a stable configuration before producing the final runs. The study was implemented with the Hugging Face Trainer using Optuna² as the optimisation backend. For each trial, the model was fine-tuned and evaluated on the validation split, and macro-F1 was used as the optimisation target. In total, 15 trials were run. The search space covered the main training hyperparameters that most strongly affect performance and convergence. The best hyperparameter configuration selected by this procedure is summarised in Table 7.

The multilingual LLMs were fine-tuned exclusively on the multilingual merged dataset described in Section 3. Training instances were formatted using an instruction-style prompt. The base prompt was: *“Eres un asistente lingüístico. Clasifica si el uso del término LGBTQ+ es reapropiativo (1) o no (0). No des la etiqueta en el texto, solo aprende del patrón.”*

For each training instance, the prompt was dynamically translated to match the language of the input text, while preserving the same instructional structure and semantics across languages. This approach ensures cross-lingual consistency and frames the task as supervised instruction tuning, where the model is fine-tuned to learn a direct mapping between the input text and the binary classification label indicating reappropriative or non-reappropriative usage.

In contrast, language-specific fine-tuning was only applied in the Spanish setting, where a Spanish-oriented LLM was trained using the Spanish monolingual dataset and a Spanish prompt. No language-specific LLMs were trained for English or Italian.

4.2.2. Run 2: Multilingual Fine-tuning with XLM-RoBERTa

For the second run submitted for each subtask, a Transformer-based classifier was trained by fine-tuning XLM-RoBERTa-base [10] on the multilingual merged dataset only. Using a single multilingual training set was a deliberate choice. First, the task definition is identical across languages, and a shared training pool increases the diversity of surface forms (slurs, self-labels, and figurative uses) that the model is exposed to. Secondly, a unified multilingual model avoids maintaining separate language-specific fine-tuned checkpoints and encourages cross-lingual transfer, where patterns learned in one language can support predictions in another.

XLM-RoBERTa-base was selected because it is a widely used multilingual encoder with strong performance on cross-lingual text classification and stable fine-tuning behaviour across many languages, including Italian, Spanish, and English. Its pretraining on large-scale multilingual corpora provides robust shared representations, which makes it well suited for the single-model, multilingual training setup adopted in this run. The hyperparameters were kept fixed to ensure comparability between models, using a *learning rate* of 3e-5, *batch size* of 16 and training for 10 epochs with *Early Stopping*. These values were chosen because they are widely used and have been shown to yield stable performance when fine-tuning Transformer-based models for text classification [4, 11].

²<https://optuna.org/>

5. Evaluation Measures

Models were assessed on the validation data using standard metrics for binary classification derived from the confusion matrix [12]. In particular, *precision*, *recall*, *accuracy*, and *F1-score* were computed. In addition, the *Area Under the ROC Curve (AUC)* was reported as a threshold-independent measure of how well the system separates the two classes [13]. Among these metrics, *accuracy*, *F1-score*, and *AUC* were taken as the main indicators of performance.

Accuracy is the simplest evaluation measure in binary classification and corresponds to the fraction of instances that are correctly predicted. Although it is easy to interpret, it can be misleading when classes are imbalanced or when the minority class is the one of interest. For this reason, it is reported together with F1-score and AUC. Accuracy is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives.

The F1-score combines precision and recall into a single value and is especially informative when the positive class is under-represented. It captures the trade-off between correctly identifying positive cases and avoiding false alarms, and is therefore well suited for hate and abuse detection tasks where errors on the minority class are critical. It is defined as:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2)$$

The AUC complements these point estimates by summarising performance across all possible decision thresholds. It measures the probability that a randomly selected positive instance is ranked above a randomly selected negative one, providing a broader view of discriminative capacity than accuracy alone [14]. Formally, it can be expressed as:

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) d(\text{FPR}) \quad (3)$$

where TPR and FPR are the true positive rate and false positive rate along the ROC curve [15].

In hate-speech-related tasks, where class distributions are typically skewed and abusive intent may be expressed indirectly, relying on a single metric can hide important failure modes. Reporting F1-score together with AUC makes it possible to evaluate both the balance between precision and recall and the system’s ability to distinguish between classes under varying thresholds, while accuracy is retained as an additional reference point for overall correctness.

In addition, the final decision criterion was aligned with the official shared-task evaluation protocol. F1-score was used as the primary metric for model selection and comparison, while the remaining measures (accuracy, precision, recall, and AUC) were reported as complementary indicators to better characterise system behaviour.

6. Results

This section reports the experimental results obtained for Task 1 across the three language subtasks. Results are presented in two parts, reflecting the two stages of the workflow. First, we summarise the scores observed during the training phase, where models were developed and compared using the balanced datasets and the train/validation split described in the previous sections. Next, we present the evaluation-phase results, corresponding to the official runs submitted to the shared task and their scores on the hidden test sets.

Run	Model / Training setup	Accuracy	Macro-F1	AUC
Run 1 (LLM + QLoRA)	OpenHermes-2.5-Mistral-7B (Spanish)	0.7483	0.7475	0.7483
	Qwen2.5-0.5B (Multilingual)	0.7483	0.7454	0.7483
Run 2 (Transformer)	XLM-RoBERTa-base (Multilingual)	0.9117	0.9115	0.9100

Table 8

Training-phase results on the validation split.

Subtask	Run 1 (LLM)			Run 2 (Transformer)		
	Macro-P	Macro-R	Macro-F1	Macro-P	Macro-R	Macro-F1
A1 (Italian)	0.6231	0.6873	0.6202	0.8627	0.8276	0.8435
A2 (Spanish)	0.5014	0.5022	0.4892	0.7442	0.6887	0.7105
A3 (English)	0.5042	0.5082	0.4945	0.5739	0.5681	0.5708

Table 9

Official evaluation results on the shared-task test sets for both runs (macro-averaged precision, recall, and F1).

Subtask	Run 1 (LLM)						Run 2 (Transformer)					
	P ₀	R ₀	F1 ₀	P ₁	R ₁	F1 ₁	P ₀	R ₀	F1 ₀	P ₁	R ₁	F1 ₁
A1 (Italian)	0.9213	0.8589	0.8890	0.3249	0.5156	0.3980	0.9676	0.9324	0.9496	0.7577	0.7228	0.7398
A2 (Spanish)	0.9146	0.9083	0.9114	0.0882	0.0962	0.0920	0.9322	0.9309	0.9316	0.5563	0.4465	0.4954
A3 (English)	0.9204	0.9072	0.9138	0.0880	0.1091	0.0974	0.9349	0.9244	0.9296	0.2129	0.2117	0.2123

Table 10

Per-class precision (P), recall (R), and F1 for both runs on the shared-task test sets. Class 1 corresponds to reclamatory intent and class 0 to non-reclamatory intent.

6.1. Training-phase Results

In this subsection, the results obtained during the training phase are reported for the models developed for each subtask and for each run. These scores were computed on the validation split and are used to summarise how the different modelling choices performed during development, before producing the final submissions.

Table 8 summarises the results obtained during the training phase for the two submitted runs. Overall, the multilingual XLM-RoBERTa model achieves the strongest scores on the development split. For the LLM-based run, different backbones were explored: while Qwen2.5-0.5B was adopted as the default choice due to its efficiency and stable behaviour, the Spanish track was trained with OpenHermes-2.5-Mistral-7B because it yielded slightly better validation performance in our experiments, and was therefore selected for the final Spanish submission.

6.2. Evaluation-phase Results

This subsection reports the official results obtained on the shared-task test sets for the two submitted runs. For each language subtask, we report the macro-averaged precision, recall, and F1-score.

Table 9 summarises overall behaviour through macro-averaged precision, recall, and F1, making the cross-language comparison straightforward and directly reflecting the competition ranking metric. In this view, the Transformer-based run achieves the best macro-F1 in all three subtasks, with the strongest performance in Italian and Spanish and a smaller, yet consistent, improvement in English.

Table 10 helps explain these trends by reporting performance separately for the two classes. Across languages, high scores for class 0 indicate that both approaches recognise the majority *non-reclamatory* category reliably. The main source of difficulty is class 1 (*reclamatory intent*), where precision and recall are substantially lower. Importantly, the Transformer run yields a clear improvement for class 1 compared to the LLM run, especially in Italian and Spanish, which largely accounts for the gains observed in macro-F1. This confirms that overall performance in this task is driven primarily by the ability to detect the minority class, rather than by differences in handling the majority class.

These results suggest that multilingual fine-tuning with XLM-RoBERTa provides a robust solution for this task under the competition evaluation setting.

7. Discussion

7.1. Qualitative Error Analysis

To complement the quantitative results, a qualitative error analysis was conducted using the official error logs released for Italian, Spanish, and English. Each file contains 20 misclassified instances, balanced across the two labels. Although the sample is small, it is useful to spot repeated patterns and to understand why the models fail beyond what is shown by global scores. A clear trend is that most errors are linked to the minority class, i.e., reclaimed usage. This is expected in a setting where the same term can act as an insult in one context and as an in-group marker in another.

Several recurring sources of confusion were observed across languages. A first issue is metalinguistic mention. Many texts do not use the term to attack or to reclaim it, but to talk about it: users quote someone else, discuss whether the word should be considered offensive, or criticise its usage. In these cases, the models seem to react strongly to the presence of the term itself. As a result, cues that signal stance are sometimes missed, such as negation, reported speech, or explicit condemnation.

A second issue is irony and humour, which are common in reclaimed language. Users may apply the term to themselves or to friends in a playful way, often with informal tone and profanity. This mixture can be difficult to interpret automatically. Even when the text contains signals of affiliation, the model may still lean towards the non-reclamatory class because aggressive wording is present.

A third source of error is topic mixing. Some messages combine identity-related terms with political debate, activism, or discussions about rights and legislation. These contexts are often polarised and emotionally loaded. The term can appear as a neutral reference, as part of a slogan, or as a weaponised label. When only the message text is available, it is not always clear which intent is dominant.

Finally, surface variation also plays a role. Creative spellings, wordplay, compounding, and morphological variants can reduce robustness, especially when these forms are rare in the training data. In practice, small changes in form may hide the lexical cues that the model has learned to rely on.

Table 11 shows representative examples (five per language) that illustrate these patterns. The same types of ambiguity appear in all subtasks, but with different emphasis. In English, many errors are linked to quotation and to mention-versus-use ambiguity. In Spanish, mistakes are often found in Pride or activism posts with dense hashtag context, where reclaimed usage is mixed with slogans and community tags. In Italian, errors include explicit in-group framing as well as politically oriented discussion, where intent is difficult to infer from text alone.

Overall, the error analysis suggests that further gains will likely require better modelling of stance and clearer separation between mentioning a term and using it directly. Targeted augmentation could also help, especially if it focuses on irony, self-reference, activism contexts, and political discourse. Under an unconstrained setting, additional context (e.g., conversational cues) could reduce ambiguity, but the text-only condition makes these cases particularly challenging.

7.2. Conclusions and Future works

This work explored two complementary approaches for identifying reclamatory intent in LGBTQ+-related terms across Italian, Spanish, and English: a QLoRA fine-tuned LLM run and a multilingual Transformer run based on XLM-RoBERTa-base. Overall, the Transformer submission achieved the strongest performance across all subtasks, while the LLM-based approach provided a competitive baseline and offered a flexible framework for rapid experimentation. Across languages, performance was mainly limited by the minority class, confirming that reclaimed usage remains difficult to detect reliably under a text-only setting.

The qualitative analysis suggests that most remaining errors are linked to ambiguous pragmatic phenomena, including metalinguistic mentions (talking about a term rather than using it), quotation

Subtask	id	Gold_label	Text
A1 (Italian)	it_169	1	chiamate ancora Louis froc...CHÉ ANCHE IO SONO UNA FROCIA DI MERDA MA NON TOCCATEMI LOUIS)
A1 (Italian)	it_1585	1	Che Libreria Adelphi sia chiusa nel weekend è un affronto a noi ricchione lavoratrici
A1 (Italian)	it_44	1	Sono in pari con #Euphoria, grazie a questa queen nei panni di una lella #Rules URL
A1 (Italian)	it_913	0	qui siamo tutti dolci e im...rché avevo bevuto 2 birre. ma chi dovrei essere? gesù cristo?
A1 (Italian)	it_489	0	Domani in Spagna verrà app...allentano l'approvazione del #DDLZan. Invidia gli spagnoli.
A2 (Spanish)	es_422	1	Hoy un poco más mariconeada, pues #OrgulloLGTBI URL
A2 (Spanish)	es_500	1	FELIZ DÍA MARICONES, y a t...basura, que os atropelle el coche de amaia #OrgulloLGTBI URL
A2 (Spanish)	es_1189	1	mariquisimo #NiQueer #NiTERF — #Orgullo2021 #Orgullo #OrgulloLGTBI URL
A2 (Spanish)	es_1199	0	Un año fuera del closet ...ER #pride #Pride2021 #Orgullo2021 #Orgullo #OrgulloLGTBI
A2 (Spanish)	es_1484	0	Hay q joderse Resulta q s... Trans y una Transgenero, yo ya lo flipo. Hágaselo mirar. URL
A3 (English)	en_1527	1	> "We do not hire faggots" ...only the retarded.
A3 (English)	en_1565	1	YES! Queer and faggot (bu... against me in a negative way and I just can't get over that.
A3 (English)	en_1106	1	I'm Canadian. I use queer... it light years away from "faggot" in terms of offensiveness.
A3 (English)	en_420	0	Ally I know My family is proud of me faggot can you say the same?
A3 (English)	en_1872	0	I've been out for a few weeks and I'm dreading the day when I get called a faggot

Table 11

Representative misclassified examples (5 per subtask) selected from the error logs.

and reported speech, ironic self-reference, and politically polarised contexts. These cases often require modelling stance and intent beyond surface lexical cues, which explains why high accuracy on the majority class can coexist with substantially lower scores for reclamatory intent.

Future improvements could follow three directions. First, training data could be enriched with more targeted examples covering the observed error patterns (e.g., pride/activism posts, ironic self-labeling, and quoted mentions), ideally with augmentation strategies that preserve intent while varying surface forms. Secondly, the models could be improved by teaching them to tell the difference between mentioning a term and using it. In many posts, the word appears because people quote it, talk about whether it is offensive, or report what someone said. In these cases, the term is present, but it is not being used with a direct intent. This could be addressed by adding extra training signals or prompt steps that make the system look for stance cues (e.g., negation, quotation, reported speech) and decide whether the speaker is condemning the term, discussing it, or using it in an in-group way before predicting the final label. Finally, under an unconstrained setting, incorporating additional context (e.g., conversational thread information or user-level signals when available) could reduce ambiguity and improve classification of borderline cases, especially for the reclamatory class.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT for grammar and spelling checks. All outputs were subsequently reviewed and edited by the authors, who take full responsibility for the content of the publication.

References

- [1] M.-C. Schreuder, Safe spaces, agency, and resistance: A metasynthesis of lgbtq language use, *Journal of LGBT Youth* 18 (2021) 256–272.
- [2] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [3] E. Sezerer, S. Tekir, A survey on neural word embeddings, 2021. URL: <https://arxiv.org/abs/2110.01804>. doi:10.48550/arXiv.2110.01804. arXiv:2110.01804.
- [4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2019)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: <https://aclanthology.org/N19-1423>. doi:10.18653/v1/N19-1423.
- [5] A. Zubiaga, Natural language processing in the era of large language models, 2024.
- [6] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, W. Pedrycz, A comprehensive survey on applications of transformers for deep learning tasks, *Expert Systems with Applications* 241 (2024) 122666.
- [7] T. Dettmers, A. Pagnoni, A. Holtzman, L. Zettlemoyer, Qlora: Efficient finetuning of quantized llms, *Advances in neural information processing systems* 36 (2023) 10088–10115.
- [8] Q. Team, et al., Qwen2 technical report, arXiv preprint arXiv:2407.10671 2 (2024).
- [9] M. Vangeli, Large language models as advanced data preprocessors: Transforming unstructured text into fine-tuning datasets, 2024.
- [10] X. Ou, H. Li, Ynu@ dravidian-codemix-fire2020: Xlm-roberta for multi-language sentiment analysis., in: *FIRE (Working Notes)*, 2020, pp. 560–565.
- [11] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
- [12] M. Grandini, E. Bagli, G. Visani, Metrics for multi-class classification: an overview, arXiv preprint arXiv:2008.05756 (2020). doi:10.48550/arXiv.2008.05756.
- [13] J. V. Carter, J. Pan, S. N. Rai, S. Galandiuk, Roc-ing along: Evaluation and interpretation of receiver operating characteristic curves, *Surgery* 159 (2016) 1638–1645. doi:10.1016/j.surg.2015.12.029.
- [14] D. Chicco, G. Jurman, The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation, *BMC genomics* 21 (2020) 6. doi:10.1186/s12864-019-6413-7.
- [15] A. P. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms, *Pattern recognition* 30 (1997) 1145–1159. doi:10.1016/S0031-3203(96)00142-2.

INFOTEC-NLP at MultiPRIDE: Region-Aware Transformer Models for Spanish Reclaimed Language Classification

Jorge Gleaves^{1,*†}, Guillermo Ruiz^{1,†}

¹INFOTEC Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación, 112 Circuito Tecnopolo Sur, Parque Industrial Tecnopolo 2, Aguascalientes, 20326, México.

Abstract

This report describes the system submitted for the MultiPRIDE Evalita 2026 Shared Task, specifically focusing on Task A, Subtask A2: Spanish Reclaimed Language Classification. [1] The task challenges participants to classify whether a term related to the LGBTQ+ context within a Spanish sentence is used with a reclaimed intent or not. This is framed as a binary classification problem. We fine-tuned two language models trained in Spanish, one limited to the Mexican Spanish and the other to Latin America. We achieved F1-scores of 0.7034 and 0.7569 on the test set, respectively. Motivation: The increasing presence of LGBTQ+ communities online has led to the appropriation of previously derogatory terms as symbols of identity and empowerment. Understanding and accurately classifying such "reclaimed language" is crucial for nuanced sentiment analysis, content moderation, and supporting inclusive online environments. It helps distinguish between hate speech and self-affirming expression, contributing to a more respectful digital discourse. Warning: This paper contains examples of explicitly offensive content.

Keywords

Reclaimed language, Spanish language processing, Binary text classification, Hate speech detection, LGBTQ+

1. Introduction

The rapid growth of user-generated content on social media platforms has intensified the need for robust Natural Language Processing (NLP) methods capable of identifying nuanced forms of meaning [2], intention, and social context. In particular, the analysis of language related to the LGBTQ+ community poses unique challenges, as certain terms historically associated with hate speech or discrimination may be deliberately reappropriated and used with self-affirming or empowering intent. Accurately distinguishing between harmful and reclaimed uses of such language is essential for applications including sentiment analysis, hate speech detection, and content moderation, where superficial lexical cues are often insufficient.

This paper presents the system developed by the team INFOTEC-NLP for participation in the MultiPRIDE Evalita 2026 Shared Task, focusing on Task A, Subtask A2, which addresses reclaimed language classification in Spanish. The task is formulated as a binary classification problem: given a Spanish sentence containing a term related to the LGBTQ+ context, the system must determine whether the term is used with a reclaiming intent or a non-reclaiming (potentially offensive or neutral) intent. The challenge lies in capturing pragmatic, cultural, and contextual signals that influence interpretation, particularly in a language as regionally diverse as Spanish.

1.1. Task description

The task aims to analyze the use of reclaimed language within the LGBTQ+ community. The goal is to determine whether certain terms —historically used as offensive or denigratory— are employed with a reclaiming or self-affirming intent. Participants are encouraged to consider:

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ jgleaves@gmail.com (J. Gleaves); luis.ruiz@infotec.mx (G. Ruiz)

id 0000-0001-7422-7011 (G. Ruiz)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

- Textual content of the message, such as arguments, slurs, derogatory terms, self-labeling, and figures of speech.
- Contextual information that can be inferred (when available), such as whether the author belongs to the LGBTQ+ community or their political orientation.

The problem is framed as a binary classification task, where systems must decide whether a term related to the LGBTQ+ context is used with reclamatory intent or non-reclamatory intent.

Task A - Textual Content focuses on systems that operate solely on the textual content of the message, without incorporating explicit user profile information. Two methodological approaches are permitted. In the *constrained approach*, participants may use only the training data provided for the task; external resources such as lexicons are allowed, but no additional annotated datasets (e.g., tweets, sentences, or corpora labeled for reclaimed language) may be incorporated. In contrast, the *unconstrained approach* permits the use of supplementary training material, including additional annotated datasets relevant to reclaimed language or other auxiliary resources. Participants choosing to work in the unconstrained setting must clearly indicate this choice when submitting their runs and provide a detailed explanation of the additional data and methods used in their accompanying technical report. **Our team selected the constrained approach.**

Subtask A2 - Spanish (Main Focus)

Participants work exclusively with Spanish texts. The objective is to identify whether LGBTQ+ related terms in a Spanish sentence are used with a reclaiming intent. The main challenge lies in capturing the pragmatic and cultural nuances of Spanish, where the same term may be offensive or reclaiming depending on context, tone, and identity-based usage.

In summary, Subtask A2 evaluates the ability of a system to understand context, intention, and semantic reclamation in Spanish, a task in which surface-level cues are often insufficient and deeper linguistic understanding is required.

To address this task, we explore the fine-tuning of two Transformer-based language models pretrained on large-scale Spanish Twitter corpora with regional awareness [3]: MEX_Large¹, specialized in Mexican Spanish, and BilmaLAT², covering multiple Latin American varieties. Both models are adapted to the task using the training data provided by the organizers, following a constrained approach, and were evaluated on the official validation and test sets.

Roadmap

The remainder of this paper is structured as follows. Section 3 describes the proposed system, including data preprocessing, tokenization strategies, model architecture, and hyperparameter optimization. Section 4 reports the experimental results obtained on the validation and test sets. Section 5 discusses the findings, highlights the main challenges observed—particularly the class imbalance and performance, performance—and outlines the directions for future work.

2. Related work

Hate speech detection is a problem that has been introduced in previous challenges, like PAN 2021 [4] which introduced three shared tasks—cross-domain authorship verification, profiling hate-speech spreaders on Twitter, and style-change detection in multi-author documents—designed to push forward reproducible research in text forensics and stylometry through new benchmark datasets and standardized evaluation. In recent years, challenges such as HOMO-MEX (2023) [5] and HOMO-LAT (2025) [6] focused on detecting LGBTQ+-related hate speech and sentiment in Spanish on Twitter and Reddit,

¹https://huggingface.co/guillermoruiz/MEX_Large

²<https://huggingface.co/guillermoruiz/BilmaLAT>

using annotated corpora and Transformer-based models that show moderate success but struggle with linguistic ambiguity and regional variation. Together, they highlight the need for more robust NLP systems based on dialect - to better classify LGBTQ+-phobic content and polarity in diverse Spanish-speaking communities.

The Transformer architecture [7] has proven to be very effective. One of the main differences with previous methods is the use of the Attention mechanism to understand the context of the words. The BERT and RoBERTa [8] models are good examples of successful models based on Transformers. The main drawback of these pretrained models is that they are focused on the English language and their performance is poor for a different language, like Spanish. This is the motivation for training models from scratch with regional information. The MEX_Large and BilmaLAT models are based on the RoBERTa architecture and were trained on a large collection of georeferenced tweets. The former was trained on messages from Mexico and the later from Latin America countries.

MEX_Large and BilmaLAT models include the option to introduce special regional tokens to guide the prediction, that is, each model has special tokens embedded; for MEX_Large, the token represents a state from Mexico's country, for BilmaLAT, the tokens represent a country from Latin America, and a time window consisting of a year and a month. See Table 1 for some tweet examples. For this challenge, the publication of the date was not necessary; therefore, we decided to fix it at _2023 _01.

For this binary classification task, the models were initialized by adding a classification head appropriate for our two target classes.

Table 1

Examples of tweets and the special tokens for each model. These are examples after the pre-processing.

Token + Tweet	Model
Coahuila _GEO Cómo estás amiga, nos conocemos? Soy soltero busco soltera. #PiedrasNegras #nava #allende #zaragoza	MEX_Large
Tamaulipas _GEO Ando de buenas que ya les devolví sus unfollows y métanselos por el culo ☺.	MEX_Large
BCS _GEO Ésa canción que cantas en silencio y la otra persona tmb. Bn raro.	MEX_Large
_do _2017 _09 Y que es lo que uno va hacer con _usr ? E ponen hoy mi Plan de dato y con 5 minutos de eso se acabó _usr _usr	BilmaLAT
_ar _2020 _03 soy demasiado buena para todo el mundo y se viven cagando en mí, que hago? me vuelvo una forra de mierda así me valoran?	BilmaLAT
_mx _2019 _12 Felicidades para la nueva pareja, y para usted también. Gracias por el pedazo de pastel. _url	BilmaLAT

3. Our system solution

Our approach was to fine-tune an early bird version of our BilmaLAT³ and MEX_Large⁴ models. The models were designed such that the where and when of a given text is captured, i.e. it learns about regions and periods of time; the model can be accessed and tested with the Huggingface framework

The core of our methodology involves fine-tuning a pre-trained multilingual Transformer model using the provided Spanish dataset. We did not use ensambles because we wanted to have a benchmark of our models. It is important to point out that the models are small (130 million parameters) and can be used or fine-tuned using personal computers or free cloud computing services like Colab.

3.1. Data Preprocessing and Cleaning

Before feeding the text into the model, a cleaning function `clean_tweet`, was applied to the text column of both the training and test datasets. This function performs the following steps:

³<https://huggingface.co/guillermoruiz/BilmaLAT>

⁴https://huggingface.co/guillermoruiz/MEX_Large

1. HTML Tag Removal: Uses BeautifulSoup to remove any HTML/XML tags.
2. User Mention Removal: Removes Twitter-style user mentions (e.g. @USER).
3. URL Removal: Removes URLs from the text.
4. Non-alphabetic Character Removal: Retains only alphabetic characters, periods, exclamation marks, and apostrophes.
5. Whitespace Normalization: Replaces multiple spaces with a single space.

3.2. Tokenization

After a complete cleaning of the data, we created a balanced dataset. The results of the models were average. On the other hand, using the data without any clean process, the results presented an improvement, therefore the tokenization is applied to the dataset without any clean process.

Tokenizer usage. Each model used its corresponding pretrained tokenizer (the RoBERTa-style tokenizers provided with BilmaLAT and MEX_Large). Text was tokenized to the model's maximum sequence length used during fine-tuning.

3.2.1. Region token selection.

- BilmaLAT: we prepended the regional token to each example. This token denotes Spain and was chosen because the dataset was assumed to contain a majority of tweets from Spain; adding provides an explicit regional signal to the multiregional model.
- MEX_Large: we prepended the regional token Mexico_City. This token was selected because Mexico City is treated as a neutral, broadly representative regional variant for Mexican Spanish idioms and thus reduces region-specific idiomatic bias.

Year/month (time) token selection.

- BilmaLAT time-token constraint: BilmaLAT restricts allowed time tokens to the range 2015–2023. To remain within this restriction and to provide a consistent temporal context, we fixed the time tokens to January 2023 (`_2023 _01`) for all examples. This choice supplies a stable “when” signal without relying on per-example metadata that was not required by the task.
- MEX_Large time tokens: no task-specific time constraint was required for MEX_Large in our setup; we used the regional token only Mexico_City and omitted varying time tokens to avoid introducing noisy temporal variation.

For each input sentence we prepended the chosen region token and the fixed time tokens (where applicable) before tokenization so the model could attend to region/time context from the first token positions.

3.3. Hyperparameter Search

To optimize the performance of the model, an automated hyperparameter search was conducted using Optuna [9]. The Trainer was re-initialized with a `model_init` function to ensure a fresh model instance for each trial. The `hp_space` function defined the search range for:

- `learning_rate`: Log-uniform distribution between 1×10^{-5} and 5×10^{-5} .
- `num_train_epochs`: Integer values between 4 and 12.
- `per_device_train_batch_size`: Categorical choices of [16, 32, 64].

Trials and evaluation; We performed 10 Optuna trials per model. Each trial:

1. instantiated a fresh model via `model_init`,
2. trained with the sampled hyperparameters and
3. evaluated in the validation split; Optuna selected the hyperparameter set with the highest validation F1-macro.

The notebook shows the `trainer.hyperparameter_search(..., n_trials=10)` call and subsequent extraction of `best_run.hyperparameters`.

Table 2

Confusion matrix on the development set for BilmaLAT.

	True	False
Label 1, Reclaimed	6	2
Label 0, Non-reclaimed	31	3

Table 3

Classification Report for the validation set using BilmaLAT

	precision	recall	f1-score	support
0	0.9118	0.9394	0.9254	33
1	0.7500	0.6667	0.7059	9
accuracy			0.8810	42
macro avg	0.8309	0.8030	0.8156	42
weighted avg	0.8771	0.8810	0.8783	42

Table 4

Confusion matrix on the development set for MEX_Large

	True	False
Label 1, Reclaimed	6	0
Label 0, Non-reclaimed	33	3

3.4. Final Model Training

After identifying the best hyperparameters, the models were re-initialized from scratch and trained using the optimal parameters found during the Optuna search. The best hyperparameters found for BilmaLAT were: learning rate of 2.6187×10^{-5} , 5 epochs and a batch size of 32. We repeated the same process for MEX_Large and the hyperparameters used were: 1.9998×10^{-5} , 8 epochs and a batch size of 32.

The final training used the provided training set (785 samples) and evaluation was performed on the official validation set (42 samples). Models were trained with standard cross-entropy loss for binary classification; class weighting or specialized loss functions were not applied in the reported runs (but are recommended to mitigate class imbalance).

4. Results

4.1. Validation Set Performance (after Hyperparameter Tuning)

After Optuna tuning we re-trained each model with the selected hyperparameters and evaluated on the official validation split (42 samples). Validation metrics reported below are the model scores obtained after the final training run described in Section 3.4 For the BilmaLAT, we got the results shown on Tables 2 and 3. The model achieved an accuracy of 88.10% and an F1-macro score of 81.56% in the validation set.

For the case of the MEX_Large model, the results are shown in Tables 4 and 5. As can be seen, the model achieved an accuracy of 92.86% and an F1-macro score of 87.83% in the validation set.

4.2. Test Set Predictions and Official Results

The final predictions on the held-out test set were produced using the re-trained models and submitted to the shared task evaluation. The official test metrics reported below are those returned by the MultiPRIDE evaluation server for each model. For BilmaLAT results are shown in Table 6 where we can see a score

Table 5

Classification Report for validation set using MEX_Large

	precision	recall	f1-score	support
0	0.9167	1.0000	0.9565	33
1	1.000	0.6667	0.8000	9
accuracy			0.9286	42
macro avg	0.9583	0.8333	0.8783	42
weighted avg	0.9345	0.9286	0.9230	42

Table 6

BilmaLAT Official results

	precision	recall	f1-score
0	0.9372	0.9014	0.9189
1	0.5420	0.6590	0.5948
macro avg	0.7396	0.78024	0.7569

Table 7

MEX_Large Official results

	precision	recall	f1-score
0	0.8948	0.9758	0.9335
1	0.7209	0.3522	0.4732
macro avg	0.8078	0.6640	0.7034

of 0.7569 of macro F1. The results for MEX_Large are shown in Table 7. It scored 0.7034 of macro F1. We can see a clear advantage of the BilmaLAT model over the MEX_Large.

5. Discussion

The performance in the validation set, with an F1-macro score of 75.69% and 70.34%, suggests that the fine-tuned BilmaLAT and MEX_Large models are reasonably effective in identifying reclaimed language in Spanish tweets, however, we can see a struggle for identification of Label 1 corresponding to reclaimed language; some examples are shown in the table 8.

Class Imbalance Impact: The class imbalance in the training data (85% Label 0, 15% Label 1) is a critical factor. Although the model shows high precision and recall for the majority class Label 0, the performance for the minority class Label 1, reclaimed language, has lower precision (0.5420) and recall (0.6590), with an F1-score of 0.5948 for BilmaLAT. This is a common challenge in imbalanced datasets, where models tend to be biased towards the majority class. However, MEX_Large seems to have more problems in identifying claims, showing a slight decrease in general.

The contrast between per-class precision and recall shows two complementary failure modes:

1. Low precision on Label 1 (BilmaLAT test) means that many predicted reclaimed examples are actually non-reclaimed;
2. Low recall on Label 1 (MEX_Large test) means that many true reclaimed examples are missed.

Both reduce macro F1 and are symptomatic of the skewed training distribution and the pragmatic difficulty of inferring reclamation from short social messages.

Hyperparameter Tuning Effectiveness: The Optuna-based hyperparameter search proved beneficial in identifying an optimal combination of learning rate, number of epochs and batch size, which likely contributed to the final validation performance.

Table 8

Examples where the model BilmaLAT selected the wrong label

id	text	Correct label
es_989	Hace dos días para vosotros la ley trans era un “retroceso en derechos” y los que estaban en la calle recibiendo palos ayer enfrentándose a la poli eran trans, maricones y bolleras; tened un poquito de vergüenza y dejad de usarnos para los argumentos que os vienen bien URL	1
es_1343	Este es el mes trending!!!! Pero pa jotear y mariconear esta todo el año y toda la vida!! #queer #pride #loveislove #LGTBI URL	1
es_1394	No importa la edad que tengas, los brazos de una madre siempre son el mejor refugio #FelizDíadeLaMadre #QueerAsFolk #FelizViernes #LovelsLove #LGBT #LGTBI #Gay #GayBlog #GayKiss #GayLove #RelatoLGBT #LGTBIQ #AmorLGBT #GayPride #AmbienteGay #Orgullo2021 #Gays #Pride2021 URL	0
es_469	Por todos los que han sido humillados desde la infancia. Agredidos o asesinados, al grito de maricón. A los que no quieren que seamos familia. A los que reivindican “terapias de conversión” A los que no quieren que se exhiba la bandera LGTBI. Visibilicemos más el #OrgulloLGTBI URL	0

Choice of Base Model: BilmaLAT and MEX_Large, being RoBERTa-based models pre-trained on a large Spanish corpus from Twitter (now X), aligns well with the task of identifying nuanced expressions in social media contexts. However, on this occasion, the task was to detect reclaimed language that is used with pride, creating an appropriation of terms for the LGBTQ+ population; this complicated the specific task and resulted in poor results in Label 1; on the other hand, it is shown that the use of special tokens for the regional selection of the Spanish language was a good choice, where BilmaLAT, adjusted for Spanish from Spain, performed better than MEX_Large with adjustments for Spanish from Mexico.

Conclusions and future work

In this work, we fine-tuned two region-aware RoBERTa models (BilmaLAT and MEX_Large) to detect reclaimed language in Spanish social posts. We achieved F1-scores of 75.69% and 70.34% in the test set, respectively. The experiments show consistent strengths on **Label 0** (non-reclaimed) and persistent weaknesses on **Label 1** (reclaimed), a pattern largely explained by the training distribution (85% Label 0, 15% Label 1). Overall, region/time tokens and Optuna tuning improved generalization, but the models still struggle to reliably identify reclaimed uses when pragmatic cues are subtle or removed by aggressive cleaning.

Future work

Our results are a clear sign of the need to have dedicated language models that are aware of regional information. This is an opportunity area for future research on NLP. We identified some improvements that could be applied in order to enhanced performance:

- **Addressing Imbalance:** Future work could explore techniques like oversampling the minority class (e.g., SMOTE), under sampling the majority class, using class weights during training, or employing different loss functions (e.g., focal loss) to mitigate the impact of class imbalance and potentially improve performance on the reclaimed language class.
- **Contextual Features:** The current approach is based primarily on text content. Incorporating contextual information from user profiles (as mentioned in the task description) could provide valuable signals to discern reclaimed intent.

- **Ensemble Methods:** Combining predictions from multiple models or different model architectures could lead to more robust performance.
- **Error Analysis:** A detailed error analysis on misclassified validation samples could reveal patterns and guide further model improvements or feature engineering.

Declaration on Generative AI

During the preparation of this work, we applied the Writefull’s model for grammar and spelling checks. After using these services, we reviewed and edited the content as needed and assume full responsibility for the content of the publication.

References

- [1] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] D. Rao, B. McMahan, Natural Language Processing with PyTorch Intelligent Language Applications Using Deep Learning, O'Reilly press, 2019.
- [3] G. Ruiz, R. Campos, T. Ramirez-delreal, D. Moctezuma, M. Graff, E. S. Tellez, Infotec-nlp at homo-lat 2025: Testing a novel multi-region spanish model to monitor opinion in latin american lgbtqi+ social media (2025).
- [4] J. Bevendorff, B. Chulvi, G. L. D. L. P. n. Sarracén, M. Kestemont, E. Manjavacas, I. Markov, M. Mayerl, M. Potthast, F. Rangel, P. Rosso, E. Stamatatos, B. Stein, M. Wiegmann, M. Wol-ska, E. Zangerle, Overview of pan 2021: Authorship verification, profiling hate speech spread-ers on twitter, and style change detection: Extended abstract, in: Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 – April 1, 2021, Proceedings, Part II, Springer-Verlag, Berlin, Heidelberg, 2021, p. 567–573. URL: https://doi.org/10.1007/978-3-030-72240-1_66. doi:10.1007/978-3-030-72240-1_66.
- [5] G. B.-E. y Helena Gómez-Adorno y Gerardo Sierra y Juan Vásquez y Scott Thomas Andersen y Sergio Ojeda-Trueba, Overview of homo-mex at iberlef 2023: Hate speech detection in online messages directed towards the mexican spanish speaking lgbtq+ population, Procesamiento del Lenguaje Natural 71 (2023) 361–370. URL: <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6566>.
- [6] G. B.-E. y Helena Gómez-Adorno y Sergio Ojeda-Trueba y Gerardo Sierra y Jessica Barco y Edgar Lee-Romero y Jocelyn Dunstan y Ruben Manrique, Overview of homo-lat at iberlef 2025: Human-centric polarity detection in online messages oriented to the latin american-speaking lgbtq+ population, Procesamiento del Lenguaje Natural 75 (2025) 413–424. URL: <http://journal.sepln.org/sepln/ojs/ojs/index.php/pln/article/view/6764>.
- [7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: North American Chapter of the Association for Computational Linguistics, 2019. URL: <https://api.semanticscholar.org/CorpusID:52967399>.
- [8] S. Jimenez, G. Dueñas, A. Gelbukh, C. A. Rodríguez-Díaz, S. Mancera, Automatic detection of regional words for pan-hispanic spanish on twitter, in: G. R. Simari, E. Fermé, F. Gutiérrez Segura, J. A. Rodríguez Melquiades (Eds.), Advances in Artificial Intelligence – IBERAMIA 2018, Springer International Publishing, Cham, 2018, pp. 404–416.
- [9] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2623–2631.

KIT-TIP-NLP at MultiPride: Continual Learning with Multilingual Foundation Model*

Barathi Ganesh HB, Michal Ptaszynski, Rene Melendez and Juuso Eronen

Text Information Processing Lab, Kitami Institute of Technology, Kitami, Hokkaido 090-0015, Japan

Abstract

This paper presents a multi-stage framework for detecting reclaimed slurs in multilingual social media discourse. It addresses the challenge of identifying reclamatory versus non-reclamatory usage of LGBTQ+-related slurs across English, Spanish, and Italian tweets. The framework handles three intertwined methodological challenges like data scarcity, class imbalance, and cross-linguistic variation in sentiment expression. It integrates data-driven model selection via cross-validation, semantic-preserving augmentation through back-translation, inductive transfer learning with dynamic epoch-level undersampling, and domain-specific knowledge injection via masked language modeling. Eight multilingual embedding models were evaluated systematically, with XLM-RoBERTa selected as the foundation model based on macro-averaged F1 score. Data augmentation via GPT-4o-mini back-translation to alternate languages effectively tripled the training corpus while preserving semantic content and class distribution ratios. The framework produces four final runs for the evaluation purposes where RUN 1 is inductive transfer learning with augmentation and undersampling, RUN 2 with masked language modeling pre-training, RUN 3 and RUN 4 are previous predictions refined via language-specific decision thresholds optimized via ROC analysis. Language-specific threshold refinement reveals that optimal decision boundaries vary significantly across languages. This reflects distributional differences in model confidence scores and linguistic variation in reclamatory language usage. The threshold-based optimization yields 2–5% absolute F1 improvement without requiring model retraining. The methodology is fully reproducible, with all code and experimental setup are available at <https://github.com/rbg-research/MultiPRIDE-Evalita-2026>.

Keywords

Reclamation Detection, Multilingual Foundation Models, Transfer Learning, Dynamic Undersampling, Back-Translation Augmentation, Language-Specific Decision Boundaries, LGBTQ+ Sentiment Analysis

1. Introduction

Automated content moderation systems often have difficulty in identifying hate speech and reclaimed language [1, 2]. Reclaimed language refers to historically derogatory language that has been reclaimed by marginalized groups, especially the LGBTQ+ community who reclaimed the language for self-expression and solidarity [3, 4]. If the algorithm cannot identify the context, it will either mark harmless content as toxic or miss actual hate speech. This paper will present the KIT-TIP-NLP system designed for the MultiPride task of EVALITA 2026, which aims to identify reclaimed language in English, Spanish, and Italian tweets [4].

The task presents three main obstacles. Firstly, the available data is limited. Secondly, the datasets are highly imbalanced, with far fewer examples of reclaimed usage than non reclaimed usage. Thirdly, the way speakers reclaim slurs varies by language and culture. A model that works well for English sarcasm may fail to grasp Italian cultural markers.

In order to tackle these problems, we employed a multistage framework. The first step was to assess a variety of multilingual models and proceed with XLM-RoBERTa as our basis. To tackle the problem of insufficient data, we implemented the GPT-4o-mini-based back-translation method that resulted in the training set being tripled while the original meaning was preserved. Additionally, we made use of a proactive sampling method during the training to ensure that the model does not turn a blind eye to

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

* This paper contains examples of explicitly offensive content.

✉ hbbg.jp@gmail.com (B. G. HB); michal@mail.kitami-it.ac.jp (M. Ptaszynski)

🆔 0000-0002-1150-2773 (B. G. HB); 0000-0002-1910-9183 (M. Ptaszynski); 0009-0004-2129-8747 (R. Melendez);

0000-0001-9841-3652 (J. Eronen)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

the minority class. To improve performance further, we assigned language-specific decision thresholds instead of applying one universal cutoff for all languages. This method assumes that the model has a higher level of confidence in certain languages and hence provides a way to get better accuracy without cumbersome retraining.

2. Methodology

Figure 1: Multi-stage multilingual hate-speech classification framework with four sequential runs refining performance via data-driven model selection, augmentation, hyperparameter selection, 5-fold CV, MLM adaptation, and threshold calibration. **RUN 1:** inductive transfer learning with optimal foundation model. **RUN 2:** transductive transfer learning on optimal foundation model followed by inductive transfer learning. **RUN 3:** threshold refinement on RUN 1 outputs and **RUN 4:** threshold refinement on RUN 2 outputs to handle linguistic nuances and cross-lingual calibration.

2.1. Data Analysis and Bias Detection

¹<https://multipride-evalita.github.io/>, Accessed on January 2026.

chi-square analysis was conducted to discover the statistical relationships between language and label distributions [5]. This exposes the label and language biases naturally existing in the dataset supplied for the modeling choices made afterwards.

2.2. Foundation Model Selection

This phase employed a two-stage approach. First, embedding representations were computed for all the tweets with each foundation models for assessing the capability of representating semantic and linguistic properties of the input tweet. A conventional machine learning (ML) pipeline was established in parallel to provide a comparative reference. Five-fold cross-validation was conducted across both embedding-based and conventional ML approaches to ensure robust and unbiased model selection. The cross-validation framework evaluated across multiple models and selected the optimal foundation model based on performance metrics computed across all folds for avoiding the selection bias.

2.3. Inductive Transfer Learning with Data Augmentation

The chosen foundational model went through an inductive transfer learning approach [6] for fine-tuning. A back-translation augmentation method was executed to deal with data deficiency with GPT-4o-mini being used as the translation engine [7]. Thru the supplied dataset, each tweet in the original language was assessed and translated to the other two languages in a systematic way. This allows for the generation of semantically equal, yet syntactically diverse paraphrases. This one-to-many translation operation has effectively increased the size of the training set threefold. The augmented dataset was then used to refine the base model further [8].

In order to address the issue of class imbalance in the classification task, a dynamic under-sampling strategy was used at the epoch level [9]. The sampling technique during each training epoch allowed every positive class instance in the training batch to have three negative class instances selected randomly without replacement. This ratio of 1:3 was maintained dynamically for all epochs, which helped the model learn distinguishing features irrespective of the class imbalance issue. For Each epoch, this sampling process was random, which introduced stochasticity and helped with the generalization of learning. The five-fold cross-validation process was used not only for the validation of the final accuracy of the model but also for hyper-parameter tuning. The final result obtained from this stage is labeled as RUN 1.

2.4. Domain Knowledge Integration via Masked Language Modeling

In the second experimental approach, the masked language modeling (MLM) was used on the foundation model with the aim of incorporating linguistic knowledge that is domain-specific. The MLM pretraining task is based on the task of predicting the randomly masked tokens in the sequences. The hyperparameter tuning was carried out using Optuna, which is an automated hyperparameter optimization library that uses efficient sampling and pruning algorithms. The final model that was adapted using MLM was again subjected to the same process of inductive fine-tuning as mentioned earlier in the section 2.3, which included back-translation augmentation, dynamic epoch-level under-sampling at the 1:3 positive to negative ratio, and five-fold cross-validation for fitness evaluation. The final result obtained from this phase is labeled as RUN 2.

2.5. Language-Specific Threshold Refinement

To further improve classification performance, language specific decision boundaries were derived for each language in the dataset. Confidence scores from the classifier were analyzed separately for English, Spanish, and Italian tweets. Language-specific thresholds were determined that optimally balanced precision and recall for each language through the ground truth labels. This step helps in accounting the distributional differences in the model's prediction scores. These thresholds were finally applied to reclassify predictions from both RUN 1 and RUN 2, resulting RUN 3 and RUN 4 respectively. This

refinement recognizes that optimal decision thresholds may differ across languages due to linguistic variation and also different patterns of language usage in the context of reclaimed slurs.

2.6. Evaluation Metrics and Statistical Analysis

All results were evaluated using standard metrics appropriate for imbalanced binary classification. Reported metrics includes averaging F1 across classes equally for minimizing majority class dominance (Macro-averaged F1 score), Precision and Recall with per class and language breakdowns, and Area Under the ROC Curve for threshold-independent measure (ROC-AUC).

Model selection during all stages of the experimentation mainly relied on macro-averaged F1 calculated through stratified 5-fold cross-validation. With this approach, the minority (reclamatory) class is treated equally together with the majority class. It is a reasonable step to take in the context of the shared task, in which both false positives and false negatives have semantic importance. Confidence intervals (95%) on cross-validation scores were computed using a set of samples to quantify the uncertainty in the performance estimates.

The language-specific performance breakdowns were calculated to evaluate multilingual generalization and pinpoint challenges. The optimization histories of Optuna were visualized using parameter importance plots and trial convergence curves to know which hyperparameters had the most impact on the validation F1 score and how rapidly the TPE sampler reached the high-quality solutions.

3. Experiments and Results

All the experiments were carried out on RTX4090 GPU workstation having the support of CUDA 12.6 and cuDNN 8.9. The training process utilized automatic mixed precision (AMP) turning to bfloat16 for the sake of computational efficiency while ensuring that the gradients remain stable. The tracking of experiments was achieved by logging hyperparameters, metrics, loss curves, and model artifacts for all the experimental runs. All the code was written in Python 3.10 using the standard scientific libraries.

3.1. Data Characteristics and Descriptive Analytics

The MultiPRIDE dataset contains slurs in LGBTQ+ contexts labeled as reclamatory or non-reclamatory for usage through tweets in English, Spanish, and Italian. The first exploratory analysis uncovered a major imbalance between the classes and a distribution of labels that varied by language. The full dataset frequency analysis revealed that the reclamatory class accounted for about 9-24% of the samples with a different distribution per language: English (9%), Spanish (18%), and Italian (24%). Chi-square tests [5] of independence between language and label produced χ^2 values that were above the critical thresholds ($p < 0.001$), which meant that the labels were not randomly distributed over the languages. This finding justified the subsequent implementation of language specific decision thresholds and stratified cross validation splitting to ensure fold wise class distribution consistency. The data statistics are depicted in Figure 2.

3.2. Foundation Model Selection and Baseline Evaluation

Eight multilingual embedding models were evaluated as candidates for downstream fine-tuning: multilingual-e5-large (state-of-the-art dense retrieval embeddings with 1024 dimensions) [10], bge-m3 (BGE multilingual model optimized for semantic search) [11], gte-multilingual-base (General-purpose Text Embeddings with 768 dimensions) [12], jina-embeddings-v3 (Jina’s multilingual dense representation model) [13], snowflake-arctic-embed-l-v2.0 (Snowflake’s large-scale multilingual embeddings) [14], LaBSE (Language-agnostic BERT Sentence Embeddings, 768 dimensions) [15], USE-multilingual (Universal Sentence Encoder for 16+ languages) [16], and XLM-RoBERTa-large (550M parameters, 24 layers, enhanced capacity for cross-lingual transfer) [17]. For each candidate model, dense representations were computed by extracting contextualized embeddings from the final transformer layer, yielding

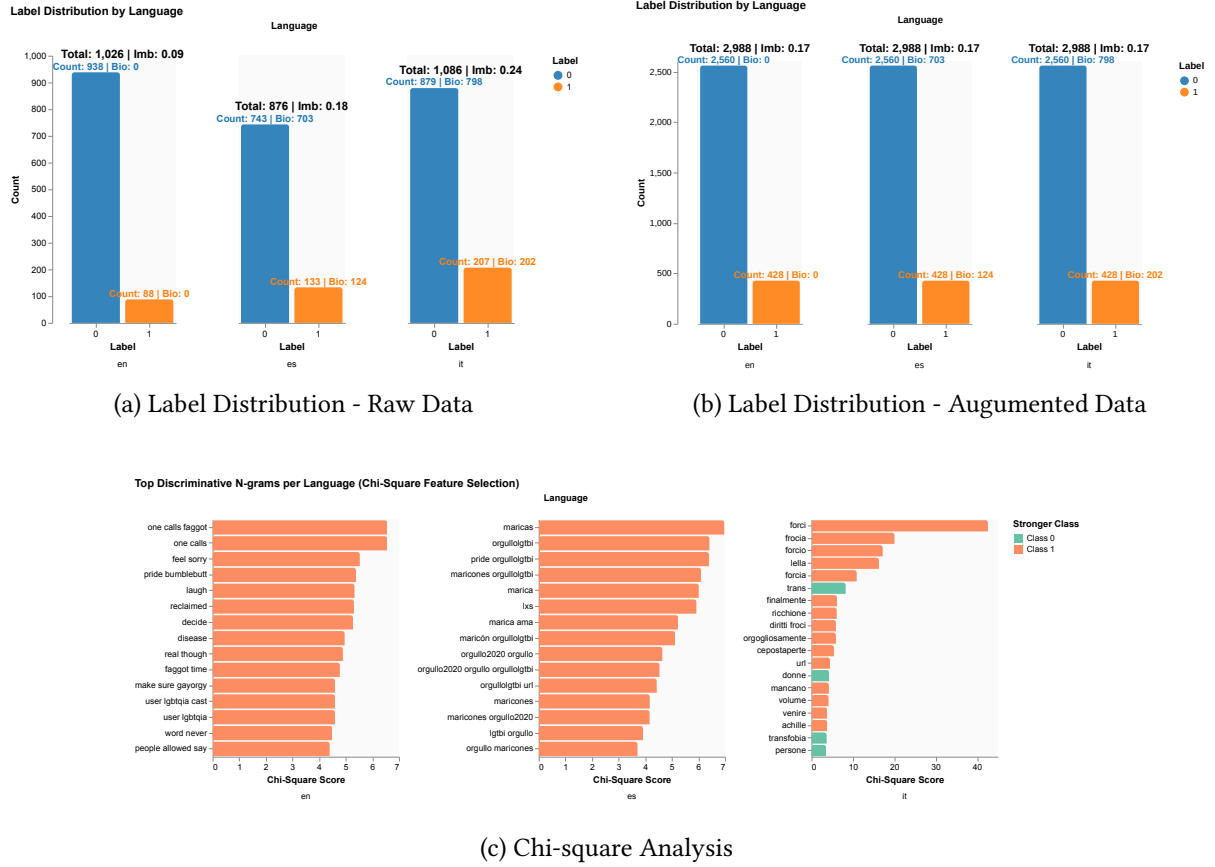


Figure 2: Data distribution statistics: label imbalance in original dataset, augmented dataset after back-translation, and chi-square analysis of language-label associations.

768-dimensional or 1024-dimensional vectors depending on the model architecture. These embeddings were subsequently used as input features for classifier training in the foundation model selection phase.

A stratified 5-fold cross-validation framework was set up, which kept the same distribution of classes in the folds (80% training, 20% validation for each fold). Conventional machine learning baselines were then trained using the computed embeddings within each fold, applying linear Support Vector Machine (Linear SVC with $C=1.0$, $\text{dual}=\text{False}$ for the sake of speed). The evaluation metrics computed for all folds include macro-averaged Accuracy, Precision, Recall, and F1 score. Besides, model selection preferred the macro-averaged F1 stability (low fold-wise variance) over the maximum absolute performance which means that the model is robust to data distribution shifts and can generalize. The analysis in Table 1 showed that XLM-RoBERTa-large managed to find the best point between the performance (macro F1 = 0.76 ± 0.04 across folds) and the computational efficiency, thus its election as the master model for all coming experiments was justified.

Table 1

Performance of Linear SVC across embedding models on the multilingual unified dataset.

Embedding model	Accuracy	Precision	Recall	F1
multilingual-e5-large	0.8018	0.7049	0.8018	0.7330
bge-m3	0.7789	0.7002	0.7789	0.7247
gte-multilingual-base	0.7765	0.6869	0.7765	0.7120
jina-embeddings-v3	0.7795	0.6860	0.7795	0.7103
snowflake-arctic-embed-l-v2.0	0.8022	0.7085	0.8022	0.7367
labse	0.7602	0.6661	0.7602	0.6879
use-multilingual	0.7603	0.6800	0.7603	0.7023
xlm-roberta-large	0.7178	0.8280	0.7178	0.7553

3.3. Inductive Transfer Learning with Data Augmentation (RUN 1)

To mitigate data scarcity, a defined back-translation augmentation strategy was implemented. Each tweet in its original language was translated to the two alternate languages using the OpenAI GPT-4o-mini API (model="gpt-4o-mini", temperature=0.0 for deterministic output, top_p=1.0). The translation prompt explicitly requested that semantic content be preserved while acknowledging natural language variation. Each original tweet generated three variants that includes native language, and two back-translated from alternate languages. This one-to-many translation schema tripled the effective training corpus size from N to approximately $3N$ samples. At the same time the reclamatory class maintaining its original 9-24% distribution (i.e. as shown in Figure 2b the class imbalance ratio was preserved during augmentation).

The RoBERTa variant² taken as a foundation model and finetuned using a custom training loop implementing dynamic undersampling at the epoch level. A custom Batch Sampler class was implemented with the iterative method reconstructed at each epoch to enforce dynamic sampling. For each training epoch, the sampler maintained a 1:3 positive-to-negative ratio by separating dataset indices into positive (label=1) and negative (label=0) groups, drawing exactly one positive sample without replacement for each batch, and drawing exactly three negative samples without replacement. This implementation ensures stochastic variation in negative sample selection across epochs, exposing the model to diverse negative examples while maintaining consistent class ratio.

Hyperparameter optimization was performed using Optuna with the Tree structured Parzen Estimator (TPE) as the underlying sampler [18]. The TPE sampler is still keeping the probabilistic model of the objective function, gradually balancing the step of exploration in the regions of hyperparameters not yet tested with the step of exploitation of the areas around the previously observed good trials that are considered as promising ones. The Optuna objective function encompassed a 5-fold stratified cross validation loop through which the hyperparameter exploration was limited to 50 trials. The hyperparameter search space included: learning rate (log-uniform, $1e-5$ to $5e-4$), batch size (categorical: 16, 32, 64), weight decay (log-uniform, $1e-5$ to $1e-2$), and dropout rate (uniform, 0.1 to 0.4). Early trial termination was implemented via Optuna’s Median Pruner with a patience threshold of 3 epochs. The trials whose validation F1 fell below the median of completed trials at the same epoch were automatically pruned. This reduces computational overhead without sacrificing final model quality. Epoch and fold level performance are shown in Figure 3.

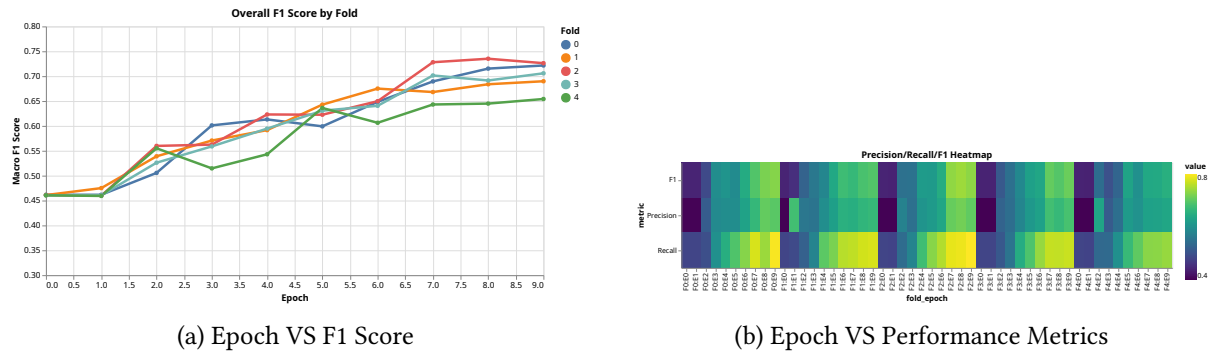


Figure 3: Fold Level Performance Metrics for Inductive Transfer Learning.

Training was done for a maximum of 10 epochs per fold with the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$) with linear learning rate warmup for the first 10% of total training steps, and linear decay to zero for the remaining steps. The loss function was weighted cross entropy, given by $L = -[w_0 \log(p_0) + w_1 \log(p_1)]$. Where w_0 and w_1 are class weights inversely proportional to class frequencies ($w_0 \approx 0.35$, $w_1 \approx 1.00$ for the imbalanced dataset). Model checkpoints were saved at each epoch based on macro-averaged F1 score on validation set. The best checkpoint for each fold was

²<https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base>, Accessed on January 2026.

chosen as the model with the highest validation F1. This setup yielded run 1, corresponding to inductive transfer learning with augmentation and dynamic undersampling.

3.4. Domain Knowledge Integration via Masked Language Modeling (RUN 2)

For incorporating domain-specific linguistic knowledge, the foundation model was subjected to a secondary MLM pretraining task before the finetuning of downstream tasks [19]. In the MLM pretraining task, 15% of the tokens in the sequence were randomly chosen using Bernoulli sampling and replaced with the [MASK] token. The model was then trained to predict the original tokens from the contextual representations. The goal of this task is to help the model learn more about the usage of LGBTQ+ discourse patterns and reclamatory language in multilingual social media settings.

The MLM pretraining phase was conducted on the augmented dataset for a maximum of 5 epochs using the AdamW optimizer with hyper parameters identified via a preliminary Optuna search. The MLM search space included parameters like learning rate (log-uniform, 1e-5 to 5e-4), batch size (categorical: 16, 32, 64), weight decay (log-uniform, 1e-5 to 1e-2), and dropout (uniform, 0.1 to 0.4). Optuna evaluated for 50 trials which optimized for minimum validation cross-entropy loss on the MLM task. Pruning was again applied via MedianPruner to terminate unpromising trials early. The MLM loss function was standard cross-entropy $L_{MLM} = -\sum \log(P_{\theta}(\text{original_token}|\text{masked_context}))$, which is summed over all masked positions in a batch. The validation loss with respect to parameter are shown in Figure 4.

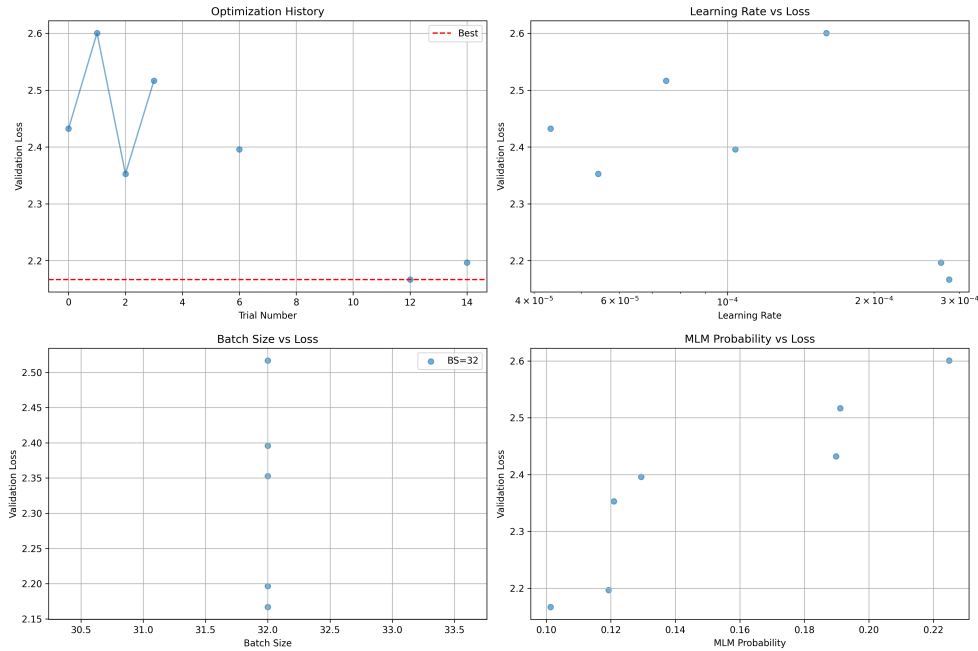


Figure 4: Transductive Transfer Learning: Parameter VS Validation Loss

Following MLM adaptation, the finetuned model was saved and subsequently used as the initialization for downstream finetuning task. This downstream finetuning pipeline was identical to run 1 where dynamic undersampling (1:3 ratio), Optuna hyperparameter optimization (50 trials, TPE sampler, MedianPruner), 5-fold stratified cross validation, and 10 epoch training with early stopping was performed. The same hyperparameter search space as run 1 was employed, allowing direct comparison of the marginal contribution of MLM adaptation. Performance metrics collected after downstream finetuning revealed the cumulative effect of both MLM adaptation and task specific optimization, resulting run 2. Empirical analysis comparing run 1 and run 2 validation F1 scores quantified the absolute and relative improvement attributable to domain knowledge injection via MLM. The impact of inductive transfer learning on transductive model with respect to the hyper parameters are shown in Figure 5.

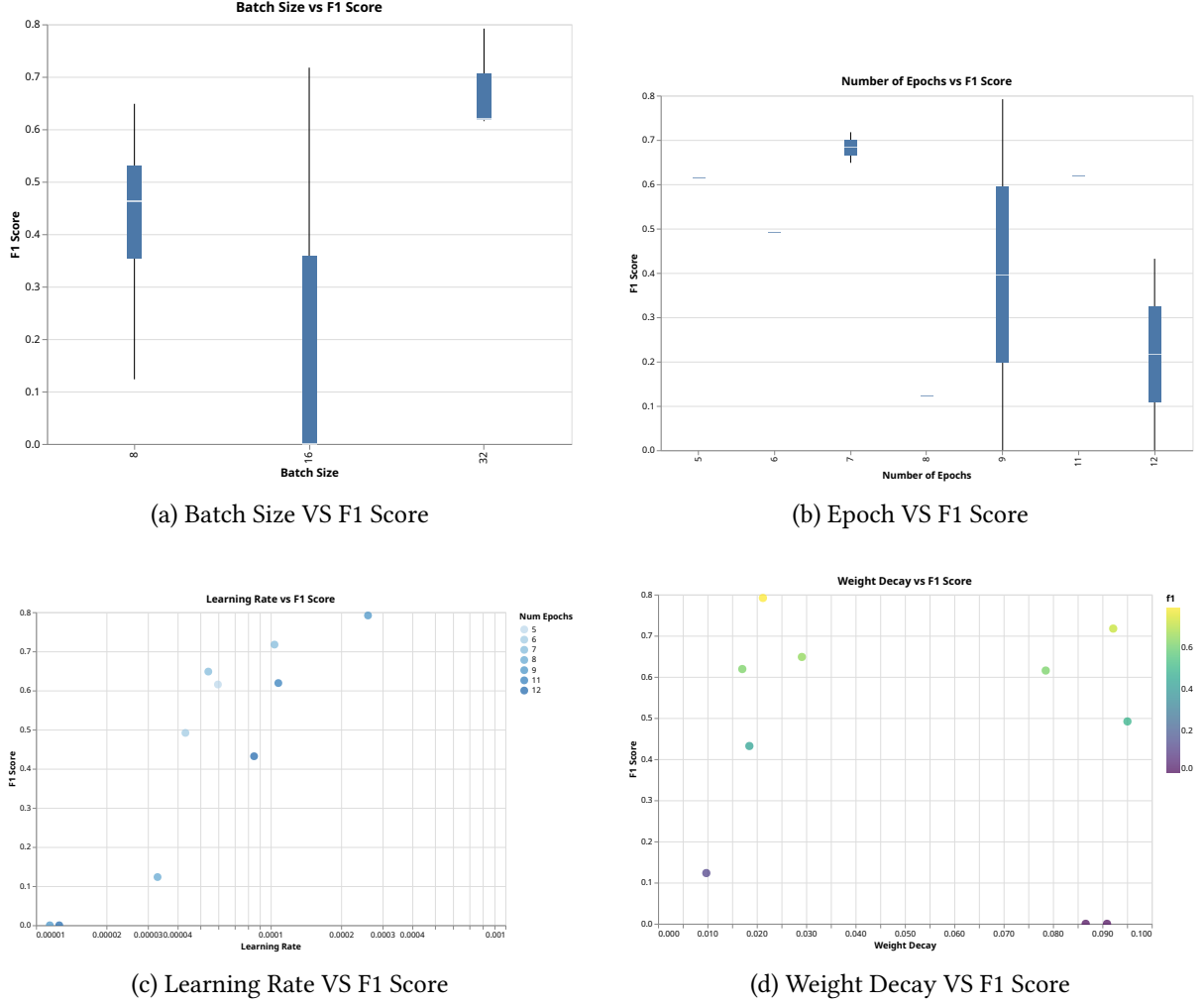


Figure 5: Inductive Transfer Learning on Transductive Model. Impact on F1 Score with respect to the parameters.

3.5. Language-Specific Threshold Refinement and Prediction Reclassification

Both run 1 and run 2 models produced continuous confidence scores via softmax normalization of the final layer logits $conf_score = \exp(\logit_1) / (\exp(\logit_0) + \exp(\logit_1))$, where $\logit_0$ and $\logit_1$ denote the class-specific logits. Default binary classification employs a threshold of 0.5 with equal misclassification costs and balanced class priors. In the MultiPRIDE task, neither assumption holds where the dataset exhibits class imbalance (9-24% minority class), and language-specific label distributions differ significantly ($\chi^2 p < 0.001$).

To optimize decision boundaries, Receiver Operating Characteristic (ROC) curve analysis was performed independently for each language. For each language (English, Spanish, Italian), validation set confidence scores were analyzed across a fine-grained threshold range. For each candidate threshold τ , performance metrics were computed includes true positive rate ($TPR = TP / (TP + FN)$), false positive rate ($FPR = FP / (FP + TN)$), and $F1score = 2TP / (2TP + FP + FN)$. The final selection for the optimal threshold τ^* of each language was based on the value that provided the maximum macro-averaged F1 score on the validation set, thus safeguarding equal performance of both the minority and the majority classes through their simultaneous consideration.

It is interesting to point out the variation of the optimal thresholds being language dependent, thus, the differences in model confidence predictions were revealed. For instance, it was found that English tweets had quite high average confidence scores for the reclamatory class which resulted in the necessity of the higher threshold (say, $\tau_{en} = 0.58$) to ensure the restriction of precision. On the

other hand, Italian tweets had lower average confidence necessitating a lower threshold (for instance, $\tau_{it} = 0.42$) to capture the instances of the minority class. Spanish thresholds were usually around these extremes ($\tau_{es} \approx 0.50$). These thresholds were subsequently enforced on the predictions so that if $conf_score \geq \tau_{language}$, the prediction will be reclamatory, otherwise, it will be non-reclamatory. The per language threshold variation has been represented in Figure 6.

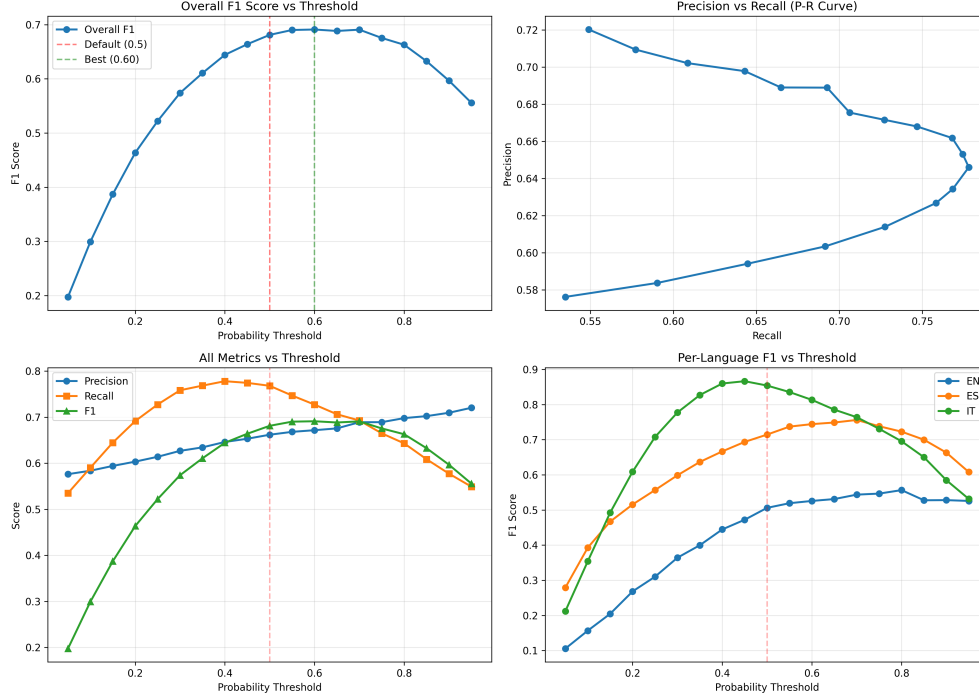


Figure 6: Threshold Analysis: Language-specific Optimal Thresholds

The predictions of run 1 model were reclassified by means of learned language specific thresholds giving rise to run 3. In the same way, reclassification of run 2 predictions was done yielding run 4. The refining of the thresholds is a very crucial post-prediction optimization step which does not require extra computational power. This step usually results in a 2-5% absolute F1 improvement by the adaptation of decision boundaries to the empirically observed confidence distributions.

4. Observations

Figure 7 presents the final submitted results on the test set. The overall results from all the runs revealed the framework’s ability to deliver considerable performance improvements in the reclamation classification task by the proper management of data scarcity, class imbalance, and multilingual variation. The foundation model selection outcomes at Table 1 revealed that RoBERTa pretrained on social media context obtains superior macro-averaged F1 stability (0.7553 ± 0.04). The recall value is also stable which is essential for minimizing missed reclamatory instances in low resource contexts. This selection decision prioritizes minority class detection over overall accuracy. This is where the concern for false negatives exceeds that of false positives in the context of LGBTQ+ discourse analysis. The back translation approach to augmentation strategy was successful in tripling the equivalent training data while maintaining the language-specific distributional properties. The maintenance of the label ratio was not trivial given the dynamic epoch level with sampling at a 1:3 positive to negative ratio. This is achieved with stochastic negative sampling selection to avoid both majority class collapse and overfitting to the same examples. The combination of weighted cross entropy loss ($w_1 \approx 1.00$ for minority class) and consistent sampling constrain achieved a stable convergence point evident in the fold-wise F1 convergence by epoch 6-7.

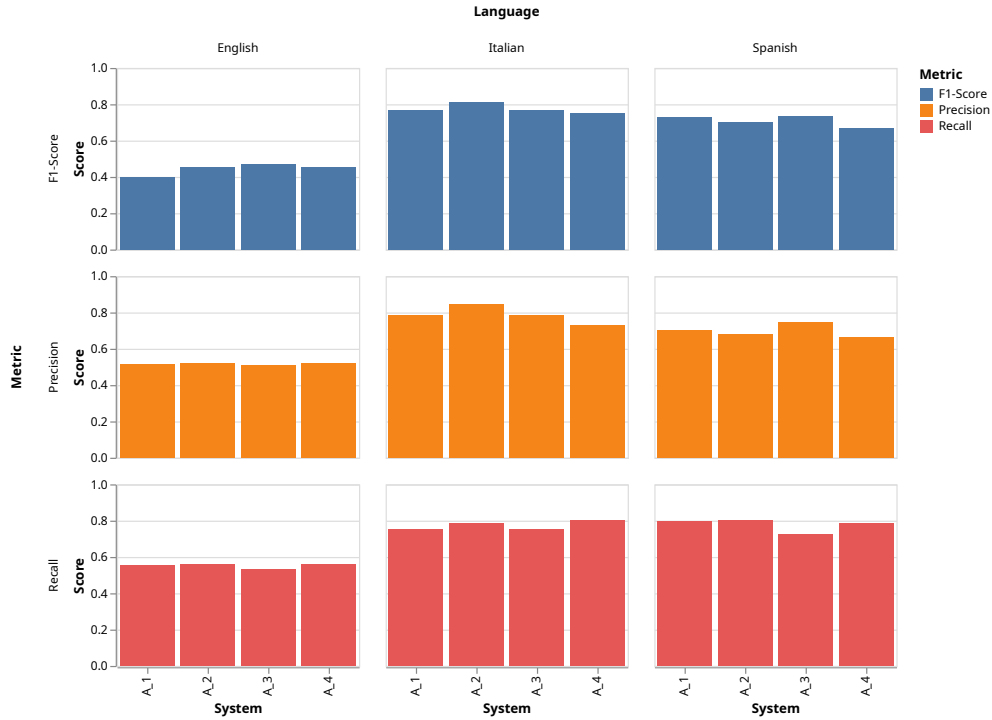


Figure 7: Final Test Set Results of Submitted Runs 1 - 4.

The integration of domain knowledge via MLM (RUN 2) hence provided language-dependent results, indicating that the adaptation of MLM is not beneficial in all multilingual settings. Although there was a marginal improvement in English, Spanish and Italian showed a more volatile response to MLM pre-training. It seems that morphologically rich languages and small amounts of pre-training data lead to different optimization spaces. The hyperparameter sensitivity analysis in Figure 5 illustrates that the adaptation of MLM models demands more accurate hyperparameter tuning, as batch size, learning rate, and weight decay all have non-monotonic relationships with F1 scores. Although the best settings lie in very narrow ranges, namely: batch size 16-32, learning rate $1e-4$ to $3e-4$, and weight decay 0.08-0.09. This highlights the significance of accurate hyperparameter tuning for domain adaptation. The Optuna TPE sampler demonstrated strong convergence properties around trial 30-35 within our 50-trial budget, validating the computational feasibility of the framework for shared task participation by avoiding grid search.

The most interpretable findings of Figure 6 reveal the threshold optimization that is specific to different languages. The decision boundaries that are chosen for reclamation detection are actually dependent on the languages: for English, it is 0.58, for Spanish, it is 0.50, and for Italian, it is 0.42. The range of difference is 16% and it is due to the diversity of languages in the context of culture-based reclamation and to the distribution of model confidence scores that are determined by the characteristics of training data. Consequently, English tweets that have more consistent reclamation markers produce higher average confidence scores, and thus, require higher thresholds to keep the precision. On the other hand, the Italian tweets that have dependent and more subtle cues in the context require lower thresholds to find the true positives. The significance in computation and the absolute F1 gains that are noticeable through this threshold refinement point out that a 0.5 default threshold assumption is a violation of linguistic universality. To conclude, by analyzing the per-language test sets, the asymmetries of the generalization patterns illustrated in the figure are revealed. The languages that have the highest native reclamatory prevalence (Italian 24%) exhibit different precision-recall dynamics compared to the low-prevalence languages such as English at 9%, where recall improvements are disproportionately benefiting the low-prevalence cases where class imbalance is most pronounced.

The analysis of the 60 incorrectly classified samples and the final test results shown in Figure 7 not

only emphasizes the linguistic and English-specific ambiguities but also suggests the social nature of language and its complexity in communication as the problem. The English category has errors primarily in the regions where the irony, hypothetical discourse, and theoretical discussion appear to blend with the reclamation of the terms used for the stigma. Samples such as "Does this mean I can start calling you a faggot?" (sarcasm asking for permission) and meta-textual discourse on the reclamation term terminology (e.g., arguing whether "faggot" could be a form of self-affirmation in various contexts) are properly classified as non-reclamatory but still, the system incorrectly classifies them as reclamatory. The problem is with the model as it fails to distinguish between the two uses that are covering "I'm gay and you're a faggot" which is delivered in a sarcastic manner and the actual reclamation. This is precisely what requires the knowledge of the speaker's intent and the social context that is more than the simple lexical word features. The Spanish language mistakes reveal different failure modes of the system like its getting stuck in the race-reclaiming in activist contexts where slurs are actually reclaimed just to take a stand against the heteronormative structures (e.g. "orgullosísimo de ser un peazo maricón" representing pride in activist framing). The Italian errors, in contrast, are mainly negative false where real reclamatory instances go unnoticed. Such cases contain the subtle cultural markers of reclamation where the intimacy shown through slur usage among in-group members is expressed (e.g. "sono pure ricchione. Bella sis" as solidarity acknowledgment). Likewise, the humor and community bonding through slur appropriation (e.g. "i viaggi ricchioni" as casual community reference) and self-descriptive pride assertions (e.g. "noi ricchione lavoratrici") that lack the explicit affirmative markers the model learned to recognize.

The Italian false negatives reveal a strong correlation with low confidence scores (0.42) and this means the model is not confident in its ability to detect Italian reclamatory patterns. These language specific error modes highlight that reclamation detection in multilingual situations cannot be based on uniform lexical or statistical patterns. English requires modeling of pragmatic context in order to filter out sarcasm and hypothetical discourse. the Spanish case benefits from activist discourse framing. In the Italian case, there is a need for explicit cultural knowledge about in-group solidarity and use of intimate language. The current limitation of inputting only text raises the performance ceiling, particularly for the Italian language, in which case non-linguistic cues (tone of voice, emoji sentiment, social relationship indicators) would aid in the recognition of reclamatory intent. The system with language-specific thresholds cannot yet overcome these representational limitations. These limitations include enhancing the training data with marking annotations that distinguish sarcasm from reclamation in English, embedding discourse context modeling for activist and political purposes in Spanish, and enhancing the training data with discourse patterns of intimate communities for Italian. The errors are consistent and systematic, the first grouping being according to the languages used and the second according to the pragmatic functions. This presumes that for multilingual reclamation detection to be accurate, not only the use of language aware models but also the understanding of reclamation through a pragmatically informed and culturally grounded approach as different in linguistic communities is a necessity.

5. Conclusions

This paper put forward a multistep method for slur reclamation detection to be applied in different linguistic environments. An effective training pipeline for the MultiPride shared task was constructed by using dynamic sampling to address the class imbalance problem and by increasing the dataset with back translation techniques. Testing has shown that the XLM-RoBERTa is a highly effective foundation model but its performance varies a lot between languages, and the main reason is the cultural factor.

One significant finding is that a "universal" decision boundary is not appropriate for the multilingual sentiment analysis. Adjusting the classification thresholds for each language resulted in a 2-5% improvement in F1 scores without incurring any extra computational costs. This implies that the models do not possess the same level of certainty in all languages, with English requiring more stringent thresholds to avoid false positives and Italian requiring less stringent thresholds to identify subtle cases of reclamation. The framework illustrates that systematic treatments of data imbalance, cross-lingual

variation identification, and domain knowledge application can lead to the development of robust multilingual sentiment classifiers that are socially aware and relevant to such domains.

However, the error analysis indicates that the models based solely on text are not capable of adequately understanding the pragmatic subtleties. The model tends to misidentify sarcasm as reclamation, particularly in English, and the lack of identification of implicit solidarity in Italian. The use of domain adaptation through MLM application led to only marginal improvements, requiring complex hyperparameter optimization and additional data. Future studies should aim to include features other than text, such as emojis or user interaction graphs, to more accurately identify the social intent underlying the language.

Declaration on Generative AI

During the preparation of this work, the authors used Generative AI in order to: Grammar and spelling check. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: *Proceedings of the 16th ACM Web Science Conference*, 2024, pp. 241–249.
- [2] B. R. Chakravarthi, R. Priyadharshini, T. Durairaj, J. P. McCrae, P. Buitelaar, P. Kumaresan, R. Ponusamy, Overview of the shared task on homophobia and transphobia detection in social media comments, in: *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, 2022, pp. 369–377.
- [3] M. Popa-Wyatt, Reclamation: Taking back control of words, *Grazer Philosophische Studien* 97 (2020) 159–176.
- [4] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [5] R. J. Tallarida, R. B. Murray, Chi-square test, in: *Manual of pharmacologic calculations: with computer programs*, Springer, 1987, pp. 140–142.
- [6] S. J. Pan, Q. Yang, A survey on transfer learning. *IEEE transactions on knowledge and data engineering*, 22 (10) 1345 (2010).
- [7] OpenAI, Gpt-4o mini: advancing cost-efficient intelligence, <https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/>, 2024. Accessed: 2026-01-07.
- [8] A. Taheri, A. Zamanifar, A. Farhadi, Enhancing aspect-based sentiment analysis using data augmentation based on back-translation, *International Journal of Data Science and Analytics* 19 (2025) 491–516.
- [9] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y.-H. Lu, S.-C. Chen, et al., Dynamic sampling in convolutional neural networks for imbalanced data classification, in: *2018 IEEE conference on multimedia information processing and retrieval (MIPR)*, IEEE, 2018, pp. 112–117.
- [10] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report, *arXiv preprint arXiv:2402.05672* (2024).
- [11] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, *arXiv preprint arXiv:2402.03216* (2024).

- [12] X. Zhang, Y. Zhang, D. Long, W. Xie, Z. Dai, J. Tang, H. Lin, B. Yang, P. Xie, F. Huang, et al., mgte: Generalized long-context text representation and reranking models for multilingual text retrieval, arXiv preprint arXiv:2407.19669 (2024).
- [13] S. Sturua, I. Mohr, M. K. Akram, M. Günther, B. Wang, M. Krimmel, F. Wang, G. Mastrapas, A. Koukounas, N. Wang, et al., jina-embeddings-v3: Multilingual embeddings with task lora, arXiv preprint arXiv:2409.10173 (2024).
- [14] S. Labs, Snowflake’s arctic embed 2.0 goes multilingual, 2024. URL: <https://www.snowflake.com/en/engineering-blog/snowflake-arctic-embed-2-multilingual/>.
- [15] F. Feng, Y. Yang, D. Cer, N. Arivazhagan, W. Wang, Language-agnostic bert sentence embedding, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 878–891.
- [16] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y.-H. Sung, et al., Multilingual universal sentence encoder for semantic retrieval, in: Proceedings of the 58th annual meeting of the Association for Computational Linguistics: system demonstrations, 2020, pp. 87–94.
- [17] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 8440–8451.
- [18] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2623–2631.
- [19] S. Gururangan, A. Marasović, S. Swayamdipta, K. Lo, I. Beltagy, D. Downey, N. A. Smith, Don’t stop pretraining: Adapt language models to domains and tasks, arXiv preprint arXiv:2004.10964 (2020).

LlaNa at MultiPRIDE: A Fine-Tuning Approach with Cross-Lingual Augmentation

Alessio Mercurio^{1†}, Giorgio Talluto^{1,†}, Irene Siragusa^{1,*} and Roberto Pirrone¹

¹Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 6, Palermo, 90128, Sicily, Italy

Abstract

In this report we present the proposed approach of LlaNa team for the MultiPRIDE challenge at EVALITA 2026 campaign. We propose a robust fine-tuning pipeline based on XLM-RoBERTa for both Italian tasks A (textual) and B (textual + user bio), enhanced by a targeted cross-lingual data augmentation strategy. By translating reclamation instances from the auxiliary English and Spanish datasets into Italian, we successfully enriched the minority class, enabling the model to learn reclamation patterns more effectively. The experimental results demonstrate the effectiveness of the proposed data-centric strategy. The developed system won in subtask B1 reaching a macro F1 score equal to 0.9021, and placed at the 5th place in subtask A1 with a score equal to 0.8835.

Keywords

LGBTQ+ Slurs Reclamation, Cross-Lingual Augmentation, Language Models

1. Warning

This paper contains examples of explicitly offensive content.

2. Introduction

The proliferation of hate speech on social media platforms has driven the Natural Language Processing (NLP) community to develop increasingly sophisticated automatic detection systems. While significant progress has been made in identifying toxic content, current state-of-the-art models often struggle with the nuances of language use within marginalized communities [1]. A critical limitation of many existing systems is their over-reliance on the presence of explicit slurs as a proxy for hate speech. This approach fails to account for the linguistic phenomenon of reclamation (or re-appropriation), where derogatory terms are repurposed by the target group, particularly the LGBTQ+ community, as markers of self-identification, pride, and solidarity [2].

In accordance with the previous EVALITA campaigns, in which both generic Hate Speech Detection (HaSpeeDe) [3] and Homotransphobia Detection (HODI) [4] challenges have been proposed in the context of the Italian language, the MultiPRIDE challenge [1] aims to explore the recognizing of usage of such slurs as reclamation terms and not as offensive one. The challenge is organized in two binary tasks: task A is a Text-based Detection, while task B is Text-based Detection with Context. The corpus consists of a multilingual dataset comprising tweets in three languages and the proposed subtasks are categorized according to linguistic affiliation: Italian (subtasks A1 and B1), Spanish (subtasks A2 and B2) and English (subtask A3). Data for task B adds the biography of the user (user bio) of the given tweet, if available.

This paper describes the system developed by the LlaNa team for subtasks A1 and B1 of the MultiPRIDE challenge in the context of the EVALITA 2026 campaign [5]. Our approach is grounded in a *data-centric* philosophy: rather than relying on complex ensemble architectures, we focused on addressing the

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ alessio.mercurio@community.unipa.it (A. Mercurio); giorgio.talluto@community.unipa.it (G. Talluto); irene.siragusa02@unipa.it (I. Siragusa); roberto.pirrone@unipa.it (R. Pirrone)

ORCID 0009-0005-8434-8729 (I. Siragusa); 0000-0001-9453-510X (R. Pirrone)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

primary bottleneck of the task, the scarcity of positive reclamation samples. We propose a robust fine-tuning pipeline based on XLM-RoBERTa [6], enhanced by a targeted cross-lingual data augmentation strategy. By translating reclamation instances from the auxiliary English and Spanish datasets into Italian, we successfully enriched the minority class, enabling the model to learn reclamation patterns more effectively.

The proposed system reached the 1st place in subtasks B1 and reaches the 5th place in subtask A1, assessing the validity of the implemented approach.

This report is arranged as follows: Section 3 describes the proposed system along with data augmentation process, while obtained results and related discussion are reported in Section 4 and 5. Concluding remarks are in Section 6.

3. Description of the system

The MultiPRIDE dataset provided for subtasks A1 and B1 consists of 1,086 Italian annotated tweets for the presence of slurs used in a reclamatory manner. For the development phase, we performed a train-test split using an 80-20 ratio, resulting in 818 and 218 samples, respectively.

3.1. Data pre-processing

We adopted a conservative pre-processing strategy to preserve the linguistic nuances typical of social media communication. Although the provided dataset contained normalized tokens for user mentions @USER and URLs URL, we retained these tokens as they appear in the distribution, without removal. This decision diverges from standard approaches that strip such metadata: we hypothesize that interaction patterns (mentions) may implicitly correlate with the context of reclamation. Regarding textual content, we intentionally avoided lower-casing since it allows for a more accurate extraction of embeddings for case sensitive Language Models (LMs), such as XLM-RoBERTa [6] which is pre-trained on cased text. In addition, uppercase text conveys significant prosodic information and emotional intensity that a lower-cased normalization would destroy. Similarly, emojis were preserved in their original form, as they represent a fundamental component of online sentiment expression and may act as crucial contextual cues in the context of slur usage.

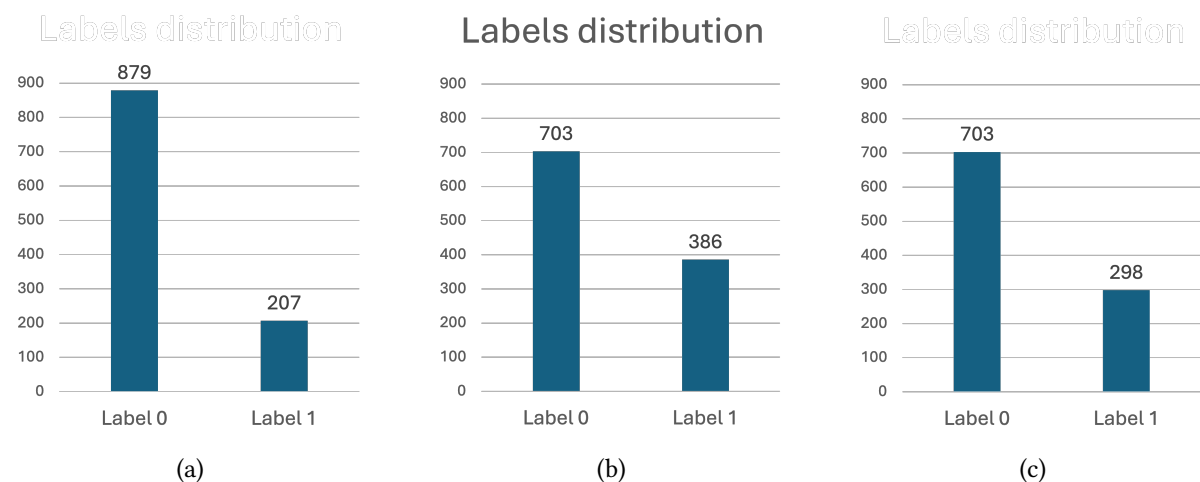


Figure 1: (a) Original unbalanced labels distribution in the Italian dataset (b) Labels distribution of the final dataset for subtask A1, using both English and Spanish dataset for augmentation (c) Labels distribution of the final dataset for subtask B1, using only the Spanish dataset for augmentation.

The primary challenge of the provided training set was the significant class imbalance (Figure 1a). The distribution featured a minority of positive samples (207 instances) compared to the negative ones (879 instances). Training deep neural networks on such skewed data typically leads to models that

overfit the majority class and fail to generalize on the phenomenon of interest. To address the data scarcity and class imbalance, we implemented a targeted cross-lingual augmentation pipeline [7], as shown in 2. Specifically, samples with a positive label, indicating the presence of reclamation, have been extracted from both English and Spanish datasets of subtasks A2, A3 and B2, and then have been machine-translated into Italian (Figure 1b). For subtask B1, the augmentation was restricted to the Spanish dataset (B2) (Figure 1c). This procedure led to the creation of two different augmented datasets for A1 and B1 tasks, where the latter had fewer positive samples to work with. Despite the resulting scarcity, the developed model will take more advantage from the additional context provided by the user bio.

For the translation process, we selected the English-Italian and Spanish-Italian models from OPUS-MT [8, 9]. The selection of these specific open-source models, as opposed to larger generative models (LLMs), was a deliberate methodological choice, as we observed that the OPUS-MT models exhibit a lower tendency to censor or euphemize offensive terms. This characteristic was crucial for our task: it ensured that the translated tweets retained the explicit slurs necessary for the model to learn reclamation patterns, rather than having them replaced with neutral synonyms or removed entirely by a safety filter.

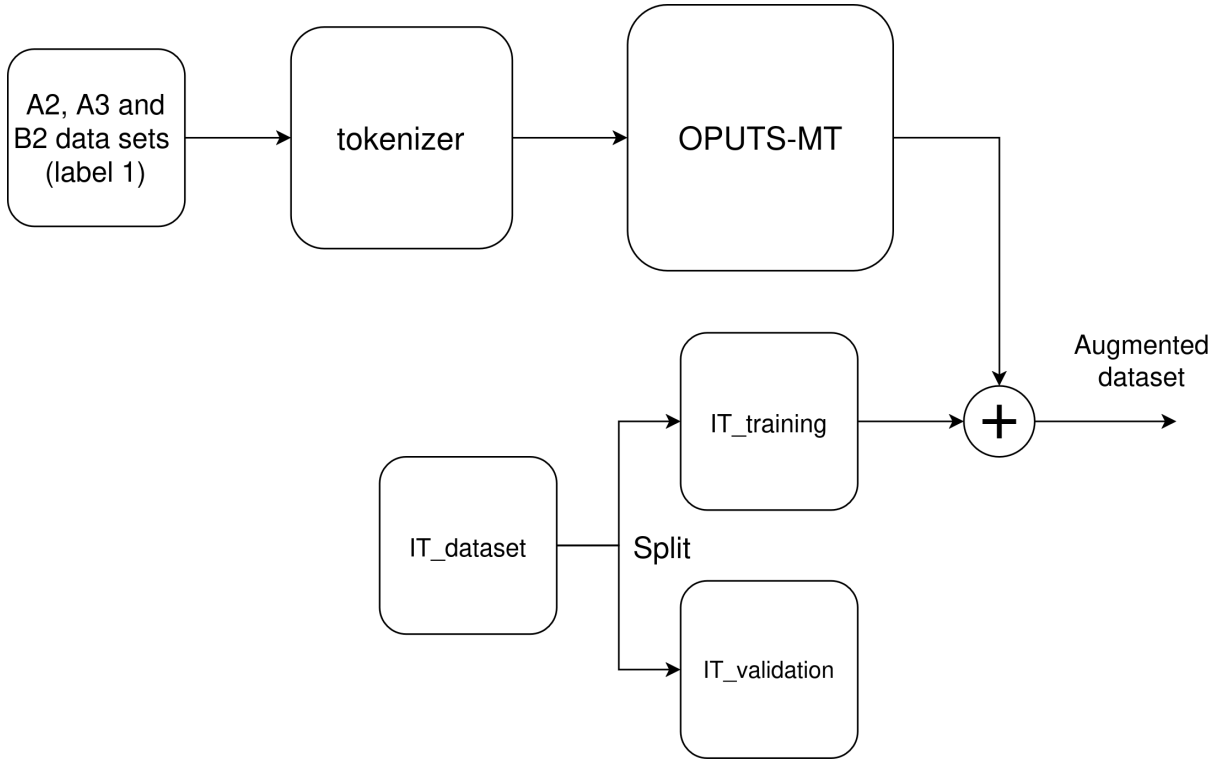


Figure 2: The proposed augmentation pipeline.

3.2. Network architecture

We employed a unique architecture for both tasks to evaluate whether the inclusion of context (user bio) yields performance improvements using the same developed model. The backbone of our system is XLM-RoBERTa-base [6, 10], selected for its state-of-the-art performance in multilingual settings and robustness on social media text. After the pre-processing phase, input sequences have been padded to a fixed length of 256 tokens. In the case of subtask B1, the input sequences are made up of the concatenation of the tweet and the user bio. The embedding of the CLS token is extracted from the last hidden layer of the encoder and fed into a custom classification head. This module consists of a dense layer projecting the 768-dimensional embedding to a hidden size of 512, followed by a normalization

layer and a ReLU activation function. To prevent overfitting, a dropout layer with a probability of 0.3 is applied before the final dense binary classifier. An overview of the developed architecture is reported in Figure 3.

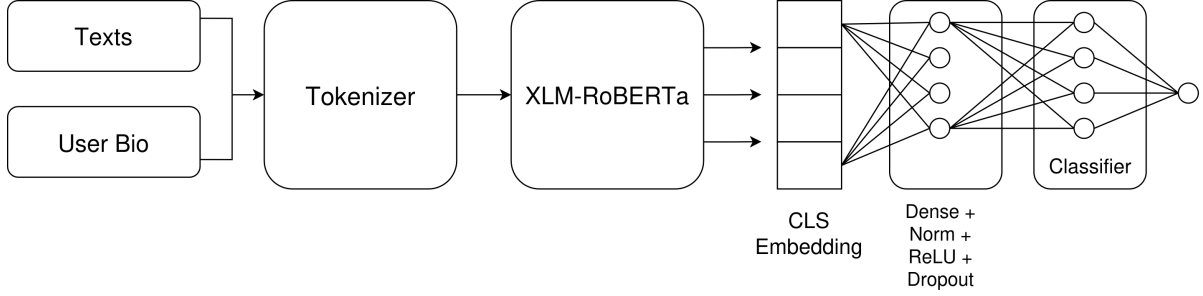


Figure 3: Schema of the proposed architecture made up of a module for embedding extraction, a dense layer with regularization followed by a binary classifier.

We trained both the LM and the dense layers on a workstation equipped with an AMD Ryzen 7 5800x CPU (32GB RAM) and an NVIDIA RTX 3060 GPU (12GB VRAM). To address the residual class imbalance, not fully resolved by the augmentation strategy, we computed and applied class weights within the loss function. The network has been optimized using AdamW [11] with a learning rate of $1e-5$ and a weight decay of 0.05. We used a batch size of 8 and a weighted Binary Cross Entropy [12] as loss function. The fine-tuning process ran for a maximum of 8 epochs, employing an early stopping mechanism with a patience of 4 epochs.

4. Results

We conducted a two-phase evaluation to assess the effectiveness of our proposed pipeline. During the development phase, we evaluate the impact of the cross-lingual augmentation on our internal validation set, while in the evaluation phase, the best developed models have been tested with the official test set.

We evaluated the performance of the developed model with our internal validation set for both subtasks A1 and B1, and obtained results are reported in Table 1. More specifically, we both tested the impact of a simple training without data augmentation (w/o augmentation) and with the proposed data augmentation strategy (w/ augmentation).

Table 1

Performances over our internal validation set (macro F1-score) for both subtasks with and without data augmentation.

Strategy	Subtask A1	Subtask B1
w/o augmentation	0.8817	0.9033
w/ augmentation	0.9131	0.9196
Δ (w/ – w/o)	+0.0314	+0.0163

The experimental results demonstrate the effectiveness of the proposed data-centric strategy, and the implemented cross-lingual augmentation pipeline proved decisive for both subtracks. We observed a performance leap when training the models with the augmented data: the macro F1-score rose from 0.8817 to 0.9131 for subtask A1, and from 0.9033 to 0.9196 for subtask B1. This confirms that transferring reclamation patterns from English and Spanish to Italian helps the model generalize better on the minority class.

The comparison between subtask A1 and subtask B1 highlights the value of contextual feature. The inclusion of the user bio in subtask B1 resulted in a further improvement which is more prominent in the w/o augmentation pipeline, where an increment of 2.4% in the macro F1-score can be found. On the contrary, the increase in the w/ augmentation pipeline is marginal (+0.7%). This suggests that

the cross-lingual augmentation provides a robust enough representation of reclamation patterns to partially compensate for the absence of explicit user profiling. Consequently, considering the superior performance across both tasks, we selected the model trained with data augmentation as our model for the blind evaluation.

The official evaluation on the held-out test set confirmed the robustness of our system, which places above the baseline proposed by the author in both subtasks. In addition, the developed system reaches the 5th place for subtask A1, while winning the competition in subtask B1. The results from Tables 2 and 3 show an macro F1-score equal to 0.8835 on subtask A1, and to 0.9021 on subtask B1, which are consistent with those previously observed during the development phase. The highest performance on subtask B1 validates our hypothesis according to which extralinguistic context like the user bio is essential for disentangling reclamation instances, even on unseen data. More specifically, as well as the training split of the dataset, the only difference in the test set from subtasks A1 to B1 is the contextual field.

Table 2

Performances over the official test set for subtask A1. The official score for the final rank is reported in bold.

Class	Precision	Recall	F1-score
0	0.9451	0.9709	0.9579
1	0.8617	0.7625	0.8091
Macro	0.9034	0.8667	0.8835
<i>Baseline</i>			<i>0.8731</i>

Table 3

Performances over the official test set for subtask B1. The official score for the final rank is reported in bold.

Class	Precision	Recall	F1-score
0	0.9624	0.9624	0.9624
1	0.8417	0.8417	0.8417
Macro	0.9021	0.9021	0.9021
<i>Baseline</i>			<i>0.8981</i>

Analyzing the class-wise performance of the positive label, we achieved a precision of 0.86 but a lower recall of 0.76 in subtask A1 (Table 2). Conversely, in subtask B1, the model achieved a more balanced performance on the minority class (Table 2). This suggests that while textual features are sufficient for high-precision detection, the user profile context is essential for recalling more subtle instances of reclamation that might otherwise be ambiguous. The diverse class imbalance among the two subtasks reinforce this assumption: since for subtask B1 we had fewer samples for the data augmentation phase and the model had fewer samples to work with, the user bio resulted crucial for performance improvement.

Finally, the system exhibited high robustness on the negative class, consistently achieving a macro F1-scores above 0.95 across both subtasks.

5. Discussion

The obtained results highlights that the primary bottleneck of our model was the lack of a sufficient samples with reclamation, rather than the architecture itself. However, despite the encouraging results, the system is not infallible. As suggested by the organizers of the task, an error analysis was performed particularly on the tweets that were misclassified by the model.

A direct inspection of the misclassified tweets reveals that errors often stem from complex contexts where the boundary between offensive usage, reclamation, and mere reporting is blurred. A common

pattern involves tweets where the slur is embedded in complex sentence structures that mix conflicting sentiments (e.g. struggle vs. pride). Consider the following example:

*“oscillo tra il volevo nascere etero e meno male che sono nato lgbt+ ed oggi più che mai grazie divinità egizie per avermi fatto r*cchione”*

In this case, the user expresses a journey from internalized conflict to self-acceptance, using slurs in a reclamatory manner. The model likely struggled to disentangle the initial negative sentiment “*volevo nascere etero*” from the final reclamation “*grazie... per avermi fatto...*”, misclassifying the intent.

A more subtle challenge is represented by tweets that contain slurs neither to offend nor to reclaim, but to report their usage by others.

*“Fedez, che voi avete decretato paladino dei diritti civili, ha ripetuto per 3 volte le parole fr*cio e n***o nelle storie Instagram, complimenti”*

In this instance, slurs are used by the author to quote their usage from a public figure and the overall tweet is a critique towards him and has an ironic tone. The classifier incorrectly flagged this as reclamation because it detected the explicit terms without grasping the metalinguistic context.

6. Conclusion

In this paper we reported the architecture proposed by the LlaNa team for MultiPRIDE subtasks A1 and B1 promoted at the EVALITA 2026 campaign. The developed system won the competition for subtask B1 and reached the 5th place subtask A1. Our experimental results demonstrate that fine-tuning a multilingual encoder (XLM-RoBERTa), when combined with a targeted cross-lingual translation data augmentation strategy, achieves robust performance in correctly classifying the use of slurs in a reclamatory manner. In particular, the comparison between the two subtasks highlights the importance to leverage extralinguistic context. The usage of the user bio, as context, resulted in a more robust classification performance, particularly in validating the generalization capabilities of developed model on unseen data. In future works we will focus on the integration of more sophisticated translation models that preserve prosodic features to better overcome issues related to data scarcity.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini 3 Pro in order to: Grammar and spelling check, Paraphrase and reword. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication’s content.

References

- [1] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] A. Galinsky, C. Wang, J. Whitson, E. Anicich, K. Hugenberg, G. Bodenhausen, The reappropriation of stigmatizing labels: The reciprocal relationship between power and self-labeling, *Psychological science* 24 (2013). doi:10.1177/0956797613482943.
- [3] C. Bosco, F. Dell’Orletta, F. Poletto, M. Sanguinetti, M. Tesconi, Overview of the EVALITA 2018 Hate Speech Detection Task, in: Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2018), CEUR-WS.org, 2018.

- [4] D. Nozza, E. Fersini, V. Patti, F. Rangel, P. Rosso, HODI at EVALITA 2023: Overview of the Homotransphobia Detection Task, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2023), CEUR-WS.org, Parma, Italy, 2023.
- [5] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [6] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, et al., Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.
- [7] T. Ranasinghe, M. Zampieri, Multilingual offensive language identification with cross-lingual embeddings, 2020. URL: <https://arxiv.org/abs/2010.05324>. arXiv:2010.05324.
- [8] J. Tiedemann, M. Aulamo, D. Bakshandaeva, M. Boggia, S.-A. Grönroos, T. Nieminen, A. Raganato, Y. Scherrer, R. Vazquez, S. Virpioja, Democratizing neural machine translation with OPUS-MT, Language Resources and Evaluation (2023) 713–755. doi:10.1007/s10579-023-09704-w.
- [9] J. Tiedemann, S. Thottingal, OPUS-MT — Building open translation services for the World, in: Proceedings of the 22nd Annual Conference of the European Association for Machine Translation (EAMT), Lisbon, Portugal, 2020.
- [10] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, et al., Huggingface’s transformers: State-of-the-art natural language processing, 2020. URL: <https://arxiv.org/abs/1910.03771>. arXiv:1910.03771.
- [11] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: <https://arxiv.org/abs/1711.05101>. arXiv:1711.05101.
- [12] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., Pytorch: An imperative style, high-performance deep learning library, 2019. URL: <https://arxiv.org/abs/1912.01703>. arXiv:1912.01703.

MilaNLP at MultiPRIDE: Evaluating Lexical, Transformer, and Retrieval-Augmented Models for Reclaimed Language

Ivana Crescenzi¹, Arianna Muti¹ and Debora Nozza¹

¹Bocconi University, Milan, Italy

Abstract

This paper describes our participation in the **MultiPRIDE** shared task at EVALITA 2026, which focuses on the detection of reclaimed language within the LGBTQ+ community. The task aims to distinguish whether a potentially offensive word is used with a *reclaimed* intent or with a *not reclaimed* one. Our system combines TF-IDF features with logistic regression, achieving strong performance through the union of word and character n -grams that capture both explicit terminology and morphological variation. We also experiment with transformer-based models (AlBERTo), instruction-tuned language models (Qwen), and retrieval-augmented approaches to investigate whether external knowledge can enhance classification. Our results show that sparse lexical features combined with linear classifiers achieve very good performance (macro F1 = 0.8959 on the official test set). On the official leaderboard, our **constrained** TF-IDF submissions ranked **2nd** in **Task A** (text-only) and **5th** in **Task B** (text+bio), while our unconstrained retrieval-augmented runs ranked 20th in Task A and 8th in Task B. This highlights that reclaimed language in Italian social media relies heavily on explicit lexical signals.

Keywords

EVALITA, Italian NLP, hate speech, reclaimed language

Content Warning: This paper contains examples of explicitly offensive content.

1. Introduction

Online social media has dramatically expanded opportunities for communication, but it has also fostered the diffusion of hate speech. In particular, speech targeting sexual orientation, gender identity and gender expression remains a serious issue for LGBTQ+ communities, affecting well-being and inclusion. Despite progress in automatic hate speech detection, many challenges persist, especially the ones that concern more subtle usage of not reclaimed terms.

A particularly complex phenomenon is *reclaimed language* which is when members of marginalized groups reappropriate slurs and offensive terms, transforming them into expressions of identity, solidarity, or resistance. This linguistic practice, also known as semantic reappropriation or requalification [1], involves taking words that have historically been used as weapons of oppression and reclaiming them as tools of empowerment. The same word, such as *frocio*, *ricchione*, or *checca* in Italian, can be deeply offensive when used by outsiders to demean and marginalize, yet serve as a powerful expression of pride and community belonging when used by LGBTQ+ individuals in self-reference or in-group communication.

This duality poses a fundamental challenge for automated content moderation and hate speech detection systems. Standard classifiers that treat offensive terms as uniformly harmful risk silencing the very communities they aim to protect, censoring expressions of identity and solidarity while failing to address genuine hate. Conversely, systems that ignore hateful uses may allow harmful content to proliferate under the guise of community language. The distinction is not merely technical but carries significant social and ethical implications for freedom of expression, community autonomy, and online safety.

We address the task of classifying whether a textual instance uses a potentially offensive term with a *reclaimed* intent or a *non-reclaimed/derogatory* intent. Detecting reclamation requires models that can

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ ivana.crescenzi@unibocconi.it (I. Crescenzi); arianna.muti@unibocconi.it (A. Muti); debora.nozza@unibocconi.it (D. Nozza)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

capture not just lexical patterns but also pragmatic context, authorial intent, and the subtle markers of in-group versus out-group language use.

The MultiPRIDE shared task frames this challenge as a binary text classification problem with two subtasks: **Task A** processes only the message text x , while **Task B** additionally incorporates the author’s user biography b when available. For each instance, systems must predict whether the text employs potentially offensive terminology with a reclaimed intent ($y=1$) or with a not reclaimed intent ($y=0$). Participants can adopt either a *constrained setting* (using only the provided training data and lexical resources) or an *unconstrained setting* (allowing external annotated datasets). This structure enables direct comparison between text-only and context-aware approaches, and between resource-limited and resource-rich scenarios.

We proceed in five steps: (i) we perform a detailed lexical and length-based analysis of the dataset, revealing systematic differences between reclaimed and non-reclaimed usage patterns, (ii) we implement a baseline that combines TF-IDF features with a linear classifier, exploring word-level and character-level representations, (iii) we consider other state-of-the-art models including transformer-based architectures and instruction-tuned language models, (iv) we explore retrieval-augmented approaches that leverage external knowledge from related hate speech corpora, and (v) we discuss error patterns and open issues specific to reclaimed language detection, providing insights into the linguistic phenomena that challenge automated classification.

Our main takeaway is that simple, interpretable models based on TF-IDF representations can achieve high performance (macro F1 = 0.896) on the test set of this task. This result is particularly noteworthy given the complexity of the phenomenon and suggests that reclaimed language in Italian social media relies heavily on explicit lexical markers that sparse feature representations can capture effectively.

2. Background

Automatic detection of hate speech has been widely studied. Early and recent work explored multi-faceted textual representations (e.g. embeddings, sentiment, topicality) for social media classification [2]. Transformer-based models and domain-specific embeddings further improve performance across different benchmarks depending on the different tasks [3]. Beyond content-only methods, user-centric and network-aware approaches analyze interaction patterns and graph structure to characterize hateful users [4]. More recently, studies have specifically targeted homophobic and LGBTQ+-oriented hate, underscoring the importance of contextual modeling for nuanced decisions [5].

The phenomenon of *abusive versus non-abusive swearing* has received particular attention in recent years. Pamungkas et al. (2020) [6] highlighted the importance of distinguishing abusive from non-abusive uses of swear words in social media, demonstrating that the same lexical form can serve different communicative functions depending on context. They developed the SWAD (Swear Words Abusiveness Dataset) corpus for English, where abusive swearing is manually annotated at the word level, and showed that lexical, syntactic, and affective features are informative for automatic prediction of swearing functions. Their work emphasized that swearing is multifaceted and often serves positive social functions, including strengthening social bonds and improving conversation harmony, and explicitly noted the importance of recognizing reclaimed uses of slurs by marginalized communities. This line of research was further developed in their 2023 study [7], which provided a more comprehensive investigation of the role of swear words in abusive language detection tasks.

Most recently, and directly relevant to our work, Draetta et al. (2024) [8] presented the first study specifically addressing slur reappropriation detection in Italian. Using Large Language Models with zero-shot prompting approaches on a filtered subset of the HODI dataset [9], they explored whether instruction-tuned models could recognize reappropriative uses without supervised training. Their work is particularly important for highlighting the linguistic and cultural specificity of reclaimed language in Italian, where terms like *frocio*, *frocia*, *ricchione*, and *checca* undergo complex semantic shifts that differ from English reclamation patterns. They developed detailed annotation guidelines distinguishing friendly contexts, political reappropriation, and artistic uses, and found that even with carefully designed

prompts, the best-performing LLM configuration achieved only macro F1 = 0.58. Their analysis revealed moderate inter-annotator disagreement (Fleiss’ Kappa = 0.57), underscoring the inherent difficulty of the task even for expert human annotators, and identified specific challenges such as the ambiguous status of *frocia* (which emerged from an already-reappropriative context) and the difficulty of determining speaker community membership from text alone.

However, the specific phenomenon of *reclaimed language* as a distinct detection task remains underexplored in mainstream datasets and systems; distinguishing self-referential reappropriation from not reclaimed usage is essential for fair and accurate moderation and analysis. The MultiPRIDE task is designed precisely to address this gap [10, 11], providing datasets in multiple languages and promoting the development of systems that can navigate this subtle distinction.

3. Data

The dataset provided consists of 1,086 Italian tweets, each associated with a binary label indicating whether the text contains reclaimed or non-reclaimed language (1 and 0, respectively). For each instance, we have access to the tweet text itself and, in approximately 92% of cases, also the author’s short biography.

Split	#Docs	Not Reclaimed	Reclaimed
Full	1,086	879	207
Train+Dev	868	703	165
Validation	218	176	42

Table 1

General statistics and stratified data split.

The dataset is imbalanced (approx. 1:4), with relatively short and homogeneous texts. Samples that use reclaimed language are shorter with more explicit slurs while not-reclaimed instances (which include both derogatory and neutral/argumentative discussions) tend to be longer and more discursive (see Appendix A, Table A).

We divide the dataset into a stratified `train_dev` split (80%) and a `validation` set (20%). Stratification preserved the 1:4 class ratio between reclaimed and not reclaimed examples. We adopt a five-fold cross-validation protocol with out-of-fold (OOF) aggregation to assess robustness and minimize variance.

4. Task Description

The MultiPRIDE task is designed as a binary text classification problem: given an Italian social media message x and, when available, a user biography b , the goal is to predict whether the text employs potentially offensive terminology with a *reclaimed* intent ($y=1$) or an *not reclaimed* intent ($y=0$). This differs from generic hate speech detection because it requires recognizing self-identification and irony. The task offers two tasks: **Task A** processes only the message text x , while **Task B** combines text x with biography b . For both tasks, participants can adopt either a *constrained* setting (using only the provided training data) or an *unconstrained* setting (allowing external datasets). We normalize all texts through a minimal pipeline: (1) lowercasing; (2) replacement of mentions and URLs with the placeholders `<USER>` and `<URL>`; and (3) whitespace normalization. We deliberately preserve punctuation and emojis since they often signal stance, irony, or emotional emphasis. For Task B (contextual setting), we concatenated biographies to tweets as `text [SEP] bio`, allowing a single model to process both sources without a separate encoder.

5. Methodology

As a first baseline, we train various linear models based on **TF-IDF** features combined with **Logistic Regression**. These approaches offer high interpretability and robustness. We test different lexical granularities, capturing both word-level cues and subword patterns. All models employ class weighting to mitigate the dataset imbalance, and their behavior can be directly interpreted in terms of the most predictive lexical features. We report detailed parameter configurations in the Appendix B.

To benchmark more expressive encoder-only architectures, we fine-tuned the Italian BERT variant **ALBERTo** (dbmdz/bert-base-italian-uncased) under two input configurations: (i) tweet text only and (ii) tweet text concatenated with the author’s biography (using [SEP] as separator).

We fine-tune ALBERTo adopting stratified 5-fold cross-validation and a class-weighted objective to mitigate imbalance. Full training hyperparameters are reported in Appendix B.

Text and biography were concatenated only when a non-empty biography was available:

$$\text{input} = \text{text} \parallel [\text{SEP}] \parallel \text{bio}.$$

We further evaluate two instruction-tuned generative models from the **Qwen Instruct** series, with 1.5B and 7B parameters, respectively. Both models were used *exclusively via prompting*, without any fine-tuning, with the goal of assessing whether general-purpose LLMs can identify reclamation phenomena through instruction following alone.

Specifically, the **Qwen Instruct 1.5B** model was employed for the *text-only* task, as this setting primarily requires linguistic understanding and can be effectively addressed by a smaller-capacity model. In contrast, the **Qwen Instruct 7B** model was adopted for the *text + bio* task, where the increased informational complexity arising from the integration of biographies demands stronger representation and reasoning capabilities, which are better supported by a larger model.

We evaluate Qwen Instruct models using a deterministic decoding setup to force binary outputs. The prompts are specified in the Appendix C and, since the tweets are in Italian, the prompts are also in Italian. Full inference settings are reported in Appendix B.

5.1. External Knowledge

To leverage external knowledge without supervised fine-tuning, we develop a retrieval-augmented pipeline using the **HODI** Subtask A corpus [9] as external knowledge base. Our approach proceeds in three stages: (i) a kNN label propagation baseline to assess direct label propagation through retrieval of nearest neighbours, (ii) a text retrieval mechanism that extracts contextual evidence rather than labels, and (iii) a classification pipeline that combines retrieved context with the original features. All texts undergo consistent preprocessing: lowercasing, entity masking (@user → <USER>, URL → <URL>), and whitespace normalization.

kNN Label Propagation Baseline. We first evaluate direct label propagation via k-Nearest Neighbors (kNN) with $k = 3$. Similarity is computed using TF-IDF cosine distance on normalized HODI texts. Each test instance receives the majority label of its top-3 neighbors, weighted by cosine similarity.

This baseline yields macro F1 = 0.55 on validation but reveals critical limitations:

- Reclaimed recall = 0.26: amplifies HODI’s class imbalance.
- Many reclaimed instances are misclassified as not reclaimed because they share lexical cues (especially slurs) with hate speech and this issue is not properly addressed by the HODI task which targets a different objective.

Text Retrieval Mechanism. We address this limitation that emerges in the kNN approach by retrieving *textual evidence* rather than labels through majority vote. For each input q , we again retrieve top- $k = 3$ HODI instances via a dual TF-IDF encoder capturing lexical and morphological patterns. Retriever design choices and parameterization are reported in Appendix B.

We pre-compute TF-IDF for all HODI instances using word 1–2 grams and character 2–5 grams. Given a test query, we build its TF-IDF vector with the same features and retrieve the top- k nearest neighbors by cosine similarity. In addition to the retrieved textual context, we incorporate similarity-based signals summarizing the composition and relevance of the retrieved neighborhood. Implementation details are reported in Appendix B. We move from direct neighbor label transfer to a supervised classifier trained on **MultiPRIDE**. For each MultiPRIDE instance, we retrieve the top- k most similar HODI texts in the same TF-IDF space. Rather than copying HODI labels, we convert the retrieved neighborhood into additional features and train a class-weighted Logistic Regression on MultiPRIDE `train_dev` labels. Concretely, the final vector concatenates: (i) TF-IDF features of the original input (tweet, or tweet+bio), (ii) TF-IDF features of a retrieved *context* obtained by concatenating (and truncating) the top- k HODI texts, and (iii) retrieval statistics such as mean/max cosine similarity and a neighborhood-composition signal. Full feature definitions and optimization details are reported in Appendix B.

6. Results

Table 2 summarizes cross-validation performance on `train_dev` (80%), while Table 3 shows results on our internal validation set (20%). Official test set results provided by the task organizers are reported in Section 6.2. For evaluation, we use **Macro F1**, Precision, and Recall, complemented by confusion matrices.

6.1. Held-out Validation Set

Linear models built on TF-IDF representations provide strong and consistent baselines. Among the variants tested, combining word- and character-level features yields the highest and most stable results. Using both tweet text and biography slightly improves the out-of-fold (OOF) performance, reaching a Macro F1 of 0.90 on average (Table 2). Notably, the purely lexical models perform remarkably well despite their simplicity.

Model	Macro F1 (mean $\pm$ std)	OOF Confusion Matrix
TEXT: Word+Char_WB	0.8853 $\pm$ 0.0266	$\begin{bmatrix} 685 & 18 \\ 40 & 125 \end{bmatrix}$
TEXT+BIO (Word+Char_WB, Char)	0.9037 $\pm$ 0.0205	$\begin{bmatrix} 693 & 10 \\ 38 & 127 \end{bmatrix}$
TEXT+BIO (word + Char_WB, Char_WB)	0.9037 $\pm$ 0.0205	$\begin{bmatrix} 693 & 10 \\ 38 & 127 \end{bmatrix}$
TEXT+BIO (Char, Char)	0.8769 $\pm$ 0.0400	$\begin{bmatrix} 690 & 13 \\ 47 & 118 \end{bmatrix}$

Table 2

Cross-validation (OOF, 5-fold) results on `train_dev`.

6.1.1. Validation Performance

Results on the validation set confirm the strong performance of lexical representations (Table 3). The text-only TF-IDF union achieves the best Macro F1 (0.9565), outperforming the text+bio variant (0.9023), which suggests that biography concatenation may introduce noise. ALBERTo reaches competitive but lower scores (0.8751 text-only, 0.8502 text+bio), indicating that the limited dataset size may constrain fine-tuning benefits. Qwen Instruct models perform significantly worse (0.1740 for 1.5B, 0.4304 for 7B), confirming that zero-shot prompting alone cannot reliably capture reclamation phenomena. Finally, retrieval variants improve over the kNN baseline (0.5461) but remain below internal TF-IDF models (0.8949 text-only, 0.8725 text+bio), suggesting partial but incomplete alignment between HODI and MultiPRIDE distributions.

Model	Macro Precision	Macro Recall	Macro F1	Confusion Matrix
TF-IDF Word(1-2)+CharWB(2-5) TEXT only	0.9488	0.9648	0.9565	$\begin{bmatrix} 172 & 4 \\ 2 & 40 \end{bmatrix}$
TF-IDF Word(1-2)+CharWB(2-5) + BIO(Char)	0.9548	0.8662	0.9023	$\begin{bmatrix} 175 & 1 \\ 11 & 31 \end{bmatrix}$
AlBERTo (TEXT)	0.9003	0.8548	0.8751	$\begin{bmatrix} 171 & 5 \\ 11 & 31 \end{bmatrix}$
AlBERTo (TEXT+BIO)	0.8788	0.8282	0.8502	$\begin{bmatrix} 170 & 6 \\ 13 & 29 \end{bmatrix}$
Qwen Instruct 1.5B (TEXT)	0.5972	0.5057	0.1740	$\begin{bmatrix} 2 & 174 \\ 0 & 42 \end{bmatrix}$
Qwen Instruct 7B (TEXT+BIO)	0.5273	0.5413	0.4304	$\begin{bmatrix} 69 & 107 \\ 13 & 29 \end{bmatrix}$
kNN Retrieval Only	0.5466	0.5457	0.5461	$\begin{bmatrix} 146 & 30 \\ 31 & 11 \end{bmatrix}$
TEXT ONLY + Retrieval	0.9025	0.8877	0.8949	$\begin{bmatrix} 170 & 6 \\ 8 & 34 \end{bmatrix}$
TEXT+BIO + Retrieval	0.9086	0.8458	0.8725	$\begin{bmatrix} 172 & 4 \\ 12 & 30 \end{bmatrix}$

Table 3
Final results on the validation set (Macro averages).

6.2. Official Test Set Results

On the official leaderboard, our **constrained** TF-IDF submissions ranked **2nd** in **Task A** (text-only) and **5th** in **Task B** (text+bio). In the **unconstrained** setting, our retrieval-augmented runs ranked **20th** in **Task A** and **8th** in **Task B**.

The evaluation reports released by the organisers include precision, recall, and F1 scores for each class. For both tasks, the constrained setting uses TF-IDF features only, while the unconstrained setting extends this configuration with a retrieval-augmented approach using the external HODI corpus.

6.2.1. Quantitative overview.

For Task A (text-only), the constrained system achieves a macro F1 of 0.8959, with very strong performance on class 0 (F1 = 0.9609) and significantly lower performance on class 1 (F1 = 0.8309). The unconstrained run shows a consistent drop in the reclaimed class (F1 = 0.7239) and in macro F1 (0.8306), suggesting that adding HODI data introduces domain mismatch effects that predominantly harm the reclaimed label. The same asymmetry holds in Task B (text+bio), although overall scores are higher. In particular, the constrained run obtains a macro F1 of 0.8826 with an excellent F1 of 0.9590 on class 0 but only 0.8063 on class 1. The unconstrained counterpart performs worse on the reclaimed class (F1 = 0.7769). Across both tasks, the model systematically achieves very high precision and recall for not reclaimed instances, typically above 0.94 for both metrics, and substantially lower recall for reclaimed instances. This indicates a pronounced bias toward the majority class, resulting in a high number of false negatives and makes reclamation errors the most critical category.

6.2.2. Impact of constrained vs. unconstrained training.

In both tasks, the constrained systems outperform the unconstrained ones, especially on the reclaimed class. While additional training data from HODI increases lexical variety, it introduces noise: many HODI examples contain explicit slurs used non-reclaimedly, shifting the decision boundary toward class 0 and decreasing the model’s sensitivity to reclaimed uses in the MultiPRIDE distribution. This is particularly evident in the recall for class 1, which drops from 0.8129 (Task A constrained) to 0.6978 (Task A unconstrained), and from 0.7338 (Task B constrained) to 0.7266 (Task B unconstrained).

Thus, constrained training appears more aligned with the annotation style and pragmatic subtleties of MultiPRIDE.

7. Analysis of Retrieval Results

As reported in Table 3, the full retrieval pipeline substantially outperforms the kNN baseline, with improved recall on the reclaimed class due to contextual disambiguation rather than biased label transfer.

To better understand how the retrieval behaves on specific instances, we examine three representative cases where the retrieval either successfully disambiguates complex pragmatic contexts or fails due to domain mismatch between the MultiPRIDE distribution and the external HODI corpus. These examples illustrate both the potential advantages of textual contextualization over simple label voting and the limitations imposed by distributional misalignment.

Semantic disambiguation : “@USER @USER @USER si sa che voi LGBT (frocì non si può dire) ragionate col culo...”

(English: “@USER @USER @USER everyone knows that you LGBT (you’re not allowed to say ‘faggots’) think with your ass...”)

This is a hateful tweet that uses the slur “frocì” while simultaneously employing irony through the parenthetical remark “frocì non si può dire” (“one cannot say faggots”), which mockingly references political correctness norms. The speaker uses the offensive term while pretending to acknowledge that it should not be used, adding a layer of sarcastic aggression. The text is correctly labeled as non-reclaimed because it deploys the slur to demean LGBTQ+ individuals, not as an act of self-identification or in-group solidarity. However, the meta-linguistic commentary (“non si può dire”) and ironic framing make this instance pragmatically complex. The pure kNN retrieval baseline identifies two out of three nearest neighbors from HODI as hateful content and one as reclaimed content, leading it to correctly predict the hateful label by majority voting. However, this decision relies on frequency patterns and does not model the ironic nuance. In contrast, the retrieval classifier, beyond considering the 67% hateful neighbors, also uses the semantic content of the 33% reclaimed neighbor to contextualize the irony and confirm the hateful label more robustly.

Corpus misalignment : “Le persone trans non sono ideologia sono esseri umani...”

(English: “Trans people are not an ideology; they are human beings...”)

This text represents a neutral rights-based discourse defending transgender people, but it does not contain reclaimed slurs, therefore it is correctly labeled as non-reclaimed. However, when the retriever searches for similar texts in the external HODI corpus, it encounters a fundamental issue: HODI is a homotransphobia detection dataset that predominantly contains hateful messages attacking LGBTQ+ individuals. The retriever matches on keywords like “trans” and “persone” (people), but the retrieved HODI examples tend to express prejudice rather than support. This introduces misleading signals: the original MultiPRIDE text is pro-LGBTQ+ and supportive, while the retrieved HODI context is anti-LGBTQ+ and hateful. The root cause is distribution mismatch: HODI’s bias toward hateful content means it cannot adequately represent MultiPRIDE’s non-reclaimed category, which includes both hate and neutral rights-based discussions. This mismatch explains why the retrieval provides limited improvement and can sometimes degrade performance.

Stylistic mismatch : “L’attivismo trans e #queer nasconde #sessuofobia...”

(English: “Trans and #queer activism hides #sexophobia...”)

The discourse originates from within the LGBTQ+ community itself and reflects internal debates about political strategies, ideological positions, or priorities. While the text discusses LGBTQ+ topics and may contain terminology associated with reclaimed language contexts, it is not using slurs in a self-referential or reappropriative manner. Instead, it expresses critique in an argumentative register that differs from external homophobic attacks. When the retriever searches for similar texts in the

HODI corpus, it often retrieves generic slur-based hate speech from hostile outsiders. The problem is that surface-level TF-IDF similarity focuses on lexical overlap (terms like “trans”, “queer”, “activism”) while missing the pragmatic distinction between intra-community debate and out-group hostility. As a result, the retrieved HODI contexts provide limited useful information and may even introduce noise. This domain mismatch is amplified by HODI’s skew toward external hate, whereas MultiPRIDE’s non-reclaimed class also includes neutral discussions, political debates, and intra-community critiques, explaining why the retrieval yields limited benefit despite its theoretical advantage over label-based kNN voting.

8. Discussion

Word and character TF-IDF (with word boundaries) capture complementary evidence: word n -grams encode explicit offensive terms and short expressions, while character n -grams capture morphological variants and orthographic creativity. Their union achieves the best trade-off between precision and recall and is remarkably stable across folds. Contrary to the initial hypothesis that biography could help out, biographies can add self-identification cues but often introduce noise: they contain hashtags, quotes, and political statements unrelated to reclamation. The observed drop from text-only to text+bio on the test set demonstrates that a naive concatenation does not generalize. Selective conditioning, or profile encoders, may leverage this information more effectively.

ALBERTo delivers consistent results and demonstrates its robustness for Italian classification, but it remains slightly below the TF-IDF baseline. The dataset’s short and lexically explicit nature favors sparse representations. Fine-tuning on limited data also risks overfitting despite class-weighted loss. As a prompt-based alternative, the Qwen Instruct models underperform significantly (Macro-F1 0.17–0.43), producing unstable outputs and invalid labels having a precision of 0.59. This highlights the difficulty of using prompt-based inference alone for context-dependent linguistic phenomena such as reclamation.

The retrieval experiments confirm that naive retrieval from external sources can complement internal features but must be carefully aligned. The kNN vote (0.5461 Macro-F1) alone is insufficient, but once contextualized, recall on reclaimed uses improves. However, external content from HODI shows domain mismatch and weak similarity (cosine scores mostly < 0.2). This partial alignment explains why retrieval method increases recall but slightly decreases overall F1 relative to the text-only baseline.

Qualitative analysis. The following qualitative analysis refers to the **official test set** released by the task organizers and is based on the errors produced by our **constrained TF-IDF (text-only) submission**. A manual inspection of the 20 misclassified tweets provided by the task organizers reveals recurrent pragmatic configurations that explain the systematic recall drop on class 1. Five tweets contain reclaimed slurs embedded in aggressive, ironic or emotionally charged exchanges, where confrontational polarity overrides reclamation cues. This pattern is observed, for instance, in tweets such as *it_326* (“non è tanto per noi froci”; “it’s not much for faggots”) and *it_913* (“urlo frocio a tutti”; “I shout faggot to everybody”), where reclaimed terms occur within heated or sarcastic discourse. Five tweets reproduce homophobic slurs by explicitly quoting them in order to criticise, parody or contextualise hate speech. Examples include *it_418*, which reports “Semo tutti froci”(“we are all faggots”), and *it_112*, where the speaker refers to offensive remarks addressed to the LGBTQ+ community. The classifier fails to detect the reported or meta-discursive stance and focuses on the surface presence of the slur. Three tweets use reclaimed identity terms in low-marked self-descriptive contexts, where the term functions as bare self-identification rather than as an explicit positive evaluation. For example, in *it_524* (“frocio lo sono pure io”; “I am also a faggot”) and *it_815* (“frocio, fluido, queer”; “faggots, fluids, queer”), reclaimed labels are used descriptively, leading the model to predict class 0. Four tweets instantiate dialogic structures that require conversational context to interpret reclamation correctly. This includes direct replies or mentions such as *it_50* (“Avete paura di non potere più picchiare e insultare i ‘frocio’?” ; “Are you scared you can no longer insult faggots and beat them up”) and *it_986* (addressing @USER), where speaker stance and pragmatic intent depend on preceding turns. Without conversation-level modelling,

the system cannot capture reclamation emerging across turns. The combination of quantitative and qualitative evidence suggests that the model is highly competent in detecting prototypical offensive uses and that reclaimed uses are more heterogeneous and pragmatically marked, making them more difficult to capture.

9. Conclusions

In this work we described our participation in the MultiPRIDE task at EVALITA 2026, focusing on the detection of reclaimed language in Italian social media. Our experiments show that simple, interpretable models based on TF-IDF features combined with linear classifiers can outperform more complex neural architectures. These results highlight the strong role of explicit lexical cues in distinguishing reclaimed from non-reclaimed uses within this domain specifically relevant to the Italian language.

The analysis further indicates that additional contextual signals, such as user biographies or external retrieved examples, do not necessarily improve performance and may introduce noise when not carefully integrated. A limitation of our comparison is that the two tasks were evaluated using different model sizes, preventing a fully controlled assessment of the biographical context’s contribution. The systematic error patterns observed across subtasks reveal the intrinsic difficulty of identifying reclaimed usage, especially in cases involving quotation, mixed-polarity contexts, or pragmatically nuanced in-group communication.

Declaration on Generative AI

Generative AI tools were used to assist with proofreading and structural suggestions during the writing of this paper. The authors take full responsibility for the final content.

References

- [1] Bianchi, C. (2014). *Slurs and appropriation: An echoic account*. Journal of Pragmatics, 66:35–44. [page:1]
- [2] Cao, R., Lee, K.-W. R., & Hoang, T.-A. (2021). *DeepHate: Hate Speech Detection via Multi-Faceted Text Representations*. arXiv:2103.11799.
- [3] Saleh, H., Alhothali, A., & Moria, K. (2021). *Detection of Hate Speech using BERT and Hate Speech Word Embedding with Deep Model*. arXiv:2111.01515.
- [4] Ribeiro, M. H., Calis, P. H., Santos, Y. A., Almeida, V. A. F., & Meira Jr., W. (2018). *Characterizing and Detecting Hateful Users on Twitter*. In Proc. ICWSM 2018, pp. 10–19.
- [5] McGiff, J., & Nikolov, N. S. (2024). *Bridging the gap in online hate speech detection: A comparative analysis of BERT and traditional models for homophobic content identification on X/Twitter*. arXiv:2405.09221.
- [6] Pamungkas, E. W., Basile, V., & Patti, V. (2020). *Do You Really Want to Hurt Me? Predicting Abusive Swearing in Social Media*. In Proc. LREC 2020, pp. 6237–6246.
- [7] Pamungkas, E. W., Basile, V., & Patti, V. (2023). *Investigating the role of swear words in abusive language detection tasks*. Language Resources and Evaluation, 57(1):155–188.
- [8] Draetta, L., Ferrando, C., Cuccarini, M., James, L., & Patti, V. (2024). *ReCLAIM Project: Exploring Italian Slurs Reappropriation Detection with LLMs*. In Proc. 5th CLiC-it 2024.
- [9] Nozza, D., Cignarella, A. T., Damo, G., Caselli, T., & Patti, V. (2023). *HODI at EVALITA 2023: Overview of the first Shared Task on Homotransphobia Detection in Italian*. In EVALITA 2023: Technical Reports. CEUR Workshop Proceedings, vol. 3473.
- [10] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, *Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task*, in: Proceedings of the Ninth Evaluation Campaign of Natural Lan-

guage Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [11] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, *Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian*, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

A. Exploratory Analysis

Text length. Across all samples, the average message length is **35 tokens** (SD 11.2) and **199 characters** (SD 61). Texts labeled as 0 are generally longer (mean **36.8** tokens; **216** characters) compared to those labeled as 1 (mean **22.5** tokens; **127** characters).

Label	Characters		Tokens	
	Mean	SD	Mean	SD
0	216.2	51.6	36.8	9.2
1	127.3	67.0	22.5	11.8

Table 4

Average text length per class (characters and tokens).

The token-length histogram shows a *mostly unimodal* distribution with a clear mode between 35–45 tokens, and a short left tail.

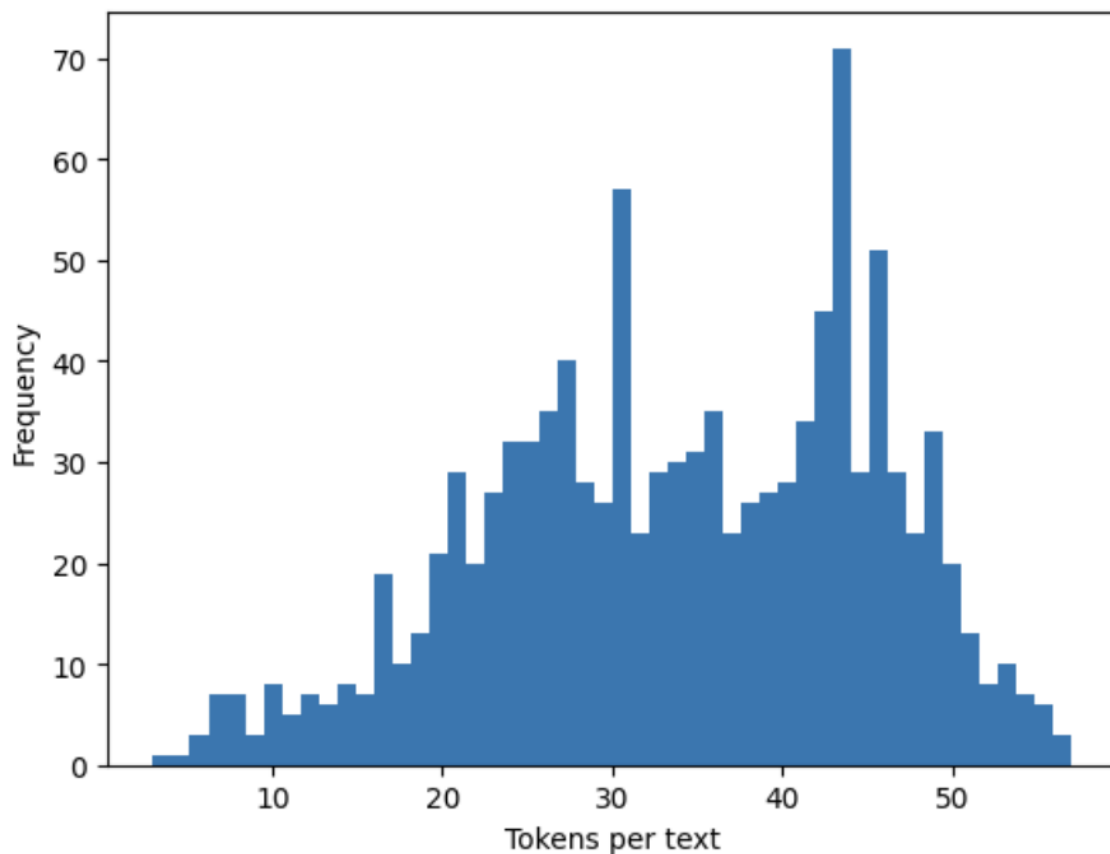


Figure 1: Distribution of token lengths across texts in the dataset.

Token frequency analysis (after stopword removal) reveals domain-coherent terms such as *trans*, *transfobia*, *donne*, *diritti*, *lgbt*, *gay*, together with placeholders like @USER and <URL>. Reclaimed samples (label=1) contain higher frequencies of commonly offensive terms (*frocio*, *froci*, *ricchione*), while offensive samples (label=0) tend to include more discursive or argumentative language (e.g., *diritti*, *transfobia*, *donne*).

Minor artifacts were detected (URLs, @USER mentions, placeholders), but no duplicated texts or IDs. A light normalization was applied: lowercasing, placeholder standardization (@USER, <URL>),

and whitespace cleaning. All experiments used a fixed random seed (42) for stratification and cross-validation.

B. Implementation Details and Hyperparameters.

B.1. TF-IDF and Linear Classifiers specifics.

Three complementary vectorizers were explored:

- **Word-level TF-IDF:** word n -grams ($n=\{1, 2\}$), sublinear term frequency, smoothed IDF, and $\text{min_df} = 2$, $\text{max_df} = 0.9$. This configuration captures both isolated slurs and short collocations.
- **Character-level TF-IDF:** two analyzers: (i) plain CHAR with n -grams of length 3–5, and (ii) CHAR_WB (word-boundary aware) with n -grams 2–5. These extract subword patterns robust to spelling variation, inflection, and creative obfuscation.
- **Feature Union:** concatenation of the Word(1–2) and Char_WB(2–5) spaces, followed by a class-weighted Logistic Regression.

B.2. ALBERTo Training Hyperparameters

The training hyperparameters defined in the official scripts were:

- **Batch size:** 16
- **Learning rate:** 2×10^{-5}
- **Epochs:** 3
- **Max sequence length:**
 - 160 for the *text-only* setting
 - 200 for the *text+bio* setting
- **Optimizer:** AdamW (HuggingFace default)
- **Seed:** 42
- **Evaluation strategy:** evaluation at every epoch
- **Loss:** dynamically weighted cross-entropy

B.3. Qwen Inference Hyperparameters

The main hyperparameters used in the inference scripts were:

- **Batch size:** 8
- **Max input length:**
 - 160 for *text-only*
 - 200 for *text+bio*
- **Max new tokens:** 2 (forcing the model to output only 0 or 1)
- **Decoding:** deterministic (`do_sample = false`, `temperature = 0.0`)
- **Few-shot:** random sampling of k labelled instances

B.4. Summary of Model Configurations

B.5. Logistic Regression Settings (Retrieval)

The classifier is trained with inverse-frequency class weighting, a regularization parameter $C = 1.0$, a maximum of 2000 iterations, and a linear optimization solver.

Model	Batch size	LR	Max length
ALBERTo (text)	16	2×10^{-5}	160
ALBERTo (text+bio)	16	2×10^{-5}	200
Qwen 1.5B (text)	8	—	160
Qwen 7B (text+bio)	8	—	200

Table 5
TF-IDF Retriever Configuration.

Component	Parameters
Word n -grams	(1-2), min_df=2, max_df=0.90, sublinear_tf
Char word-boundary	(2-5), min_df=2, sublinear_tf
Metric	Cosine similarity
Truncation	300 characters per retrieved text

B.6. Retriever Configuration

B.7. Similarity-based Features (Retrieval)

Beyond the retrieved textual context, we extract additional similarity-based signals to summarize the retrieved neighborhood. These include:

- the fraction of reclaimed instances among the top- k retrieved texts;
- the average cosine similarity between the query and retrieved instances;
- the maximum cosine similarity observed in the neighborhood.

The final representation stacks multiple views:

$$[\text{TF-IDF}(\text{text}), \text{TF-IDF}(\text{ctx_text}), [\text{ctx_pos_frac}, \text{ctx_sim_mean}, \text{ctx_sim_max}]]$$

C. Prompts Used

Prompts for Qwen Instruct 1.5B (text-only).

Prompt:

Classifica il testo come 1 se contiene un uso riappropriato di termini LGBTQ+, oppure 0 se non riappropriato. Rispondi solo con 0 o 1.

Prompt for Qwen Instruct 7B (text+bio).

Prompt:

Sei un classificatore per il task MultiPRIDE (italiano). Ricevi un testo e, se presente, la bio del profilo (separata da [SEP]). Decidi se il termine è usato con riappropriazione (1) oppure no (0). Rispondi solo con 0 oppure 1.

NamDang at MultiPride@EVALITA 2026: Multilingual Classification of Reclaimed Language in LGBTQ+ Discourse using Transformer-based Models

Nam Dang^{1,2}, Vo Tuan Kiet^{1,2} and Dang Van Thin^{1,2}

¹University of Information Technology-VNUHCM, Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

²Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This paper describes the system developed for Task A of the MultiPride shared-task at EVALITA 2026. The task focuses on the identification of reclaimed language used by the LGBTQ+ community, where potentially offensive terms are reappropriated and used as expressions of identity and pride rather than discrimination. This task is particularly challenging due to limited and imbalanced data, as well as the semantic ambiguity between reclaimed and insulting language, where similar derogatory terms may appear in both labels. To address this problem, we adopt a transfer learning approach based on the power of pre-trained multilingual language models. We fine-tune the XLM-R and mDeBERTa models on a multilingual training dataset that combines three languages: English, Italian, and Spanish. Additionally, we investigate the impact of data augmentation by comparing models trained with and without augmented data. The experimental results demonstrate that our approach achieves competitive performance compared with other baselines and methods in the overall MultiPride ranking.

Warning for Explicit Content: This paper contains examples of language that may be offensive, including slurs and derogatory terms related to the LGBTQ+ community. These examples are included solely for the purpose of scientific analysis.

Keywords

Multilingual NLP, Pre-trained Language Models, Text Classification, MultiPride

1. Introduction

The automatic detection of abusive, offensive, and discriminatory language has become an important research topic in Natural Language Processing, particularly in the context of online platforms and social media[1, 2]. Language used in these environments often reflects broader societal issues such as sexism, homophobia, and discrimination against marginalized communities, motivating the development of computational methods for identifying harmful content.

Among these challenges, language targeting the LGBTQ+ community[3] presents specific complexities. Members of the community are frequently subjected to hate speech and derogatory expressions, which can reinforce social exclusion and discrimination. At the same time, LGBTQ+ speakers may intentionally reappropriate slurs and offensive terms as reclaimed language, using them to express identity, solidarity, or pride. This phenomenon of reclamation has been widely discussed in sociolinguistic literature[4] and highlights the importance of context and speaker intent in language interpretation.

From a computational perspective, reclaimed language poses a significant challenge for text classification systems. The same lexical items may appear in both insulting and self-referential contexts, making it difficult to rely on surface-level features alone. Distinguishing between discriminatory and reclaimed uses therefore requires models to capture contextual, semantic, and pragmatic information beyond individual words[5, 6].

The MultiPride shared task at EVALITA 2026[7] directly addresses this problem by focusing on the identification of reclaimed language related to the LGBTQ+ community. In Task A, systems are required to distinguish between reclaimed and offensive uses of potentially derogatory expressions. The task

EVALITA 2026: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Bari, Italy, February 2026

✉ 24521098@gm.uit.edu.vn (N. Dang); kietvt@uit.edu.vn (V. T. Kiet); thindv@uit.edu.vn (D. V. Thin)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

involves multilingual data in English, Italian, and Spanish, and is characterized by limited training data and class imbalance, further increasing the difficulty of the classification problem.

In this paper, we present the system developed for Task A of the MultiPride shared task. Our approach is based on fine-tuning pre-trained transformer-based language models[8], which have demonstrated strong performance across a wide range of Natural Language Processing tasks. Specifically, we employ multilingual transformer models to leverage shared representations across languages and improve generalization in low-resource settings. We experiment with XLM-RoBERTa-base[9] and mDeBERTa-v3-base[10], training the models on a combined multilingual dataset and evaluating the impact of data augmentation strategies[11].

2. Methodology

The experimental framework of this study is structured around two distinct experimental runs, both utilizing the datasets described in Section 2.

- **run-1**: utilizes the augmented dataset containing synthetic examples to evaluate the impact of data expansion
- **run-2**: serves as a baseline, trained exclusively on the original MultiPride dataset without augmentation.

To address the classification task, this research employs the transformer-based architecture provided by the Hugging Face ecosystem[12]. Transformer models are particularly suited for text classification due to their self-attention mechanisms, which capture long-range contextual dependencies across diverse linguistic structures. The selection of Hugging Face transformers ensures a robust, scalable, and reproducible framework, leveraging state-of-the-art pretrained weights that have demonstrated significant efficacy in cross-lingual transfer learning.

2.1. Models

The methodology incorporates two distinct pretrained models, selected for their specific architectural strengths and suitability for the experimental objectives.

This study builds upon two pretrained Transformer-based architectures, RoBERTa[13] and DeBERTa[14], which differ primarily in how positional information is modeled within self-attention. RoBERTa follows a standard encoder-only Transformer design in which token embeddings and absolute positional embeddings are summed at the input layer and then passed through stacked self-attention and feed-forward layers. In contrast, DeBERTa introduces a disentangled attention mechanism that represents content and relative positional information separately, allowing them to interact explicitly within the attention computation. By decoupling token semantics from positional encoding, DeBERTa provides a more expressive mechanism for modeling word order and contextual relationships, motivating its inclusion alongside RoBERTa for a comparative evaluation of positional modeling strategies.

To support a controlled comparison in the multilingual setting, we evaluate XLM-RoBERTa (XLM-R)[9] and mDeBERTa-v3[10]—multilingual extensions of RoBERTa and DeBERTa—under two independent experimental runs. **run-1** is conducted on the augmented dataset, while **run-2** uses the original MultiPride dataset. In each run, the two models are trained and evaluated in parallel, and a single best-performing model is selected independently for that run based on validation performance, rather than assuming consistent superiority of one architecture across data conditions. The model chosen in **run-1** is thus the one that empirically demonstrates greater robustness when exposed to augmented data, potentially reflecting architectural properties that better accommodate increased diversity or noise. Conversely, the model selected in **run-2** is the one that more effectively captures the characteristics of the original data distribution, favoring precise contextual or positional representations. Importantly, these interpretations are grounded solely in observed validation outcomes, and architectural differences are discussed only to contextualize empirical results, without imposing prior assumptions about model advantage.

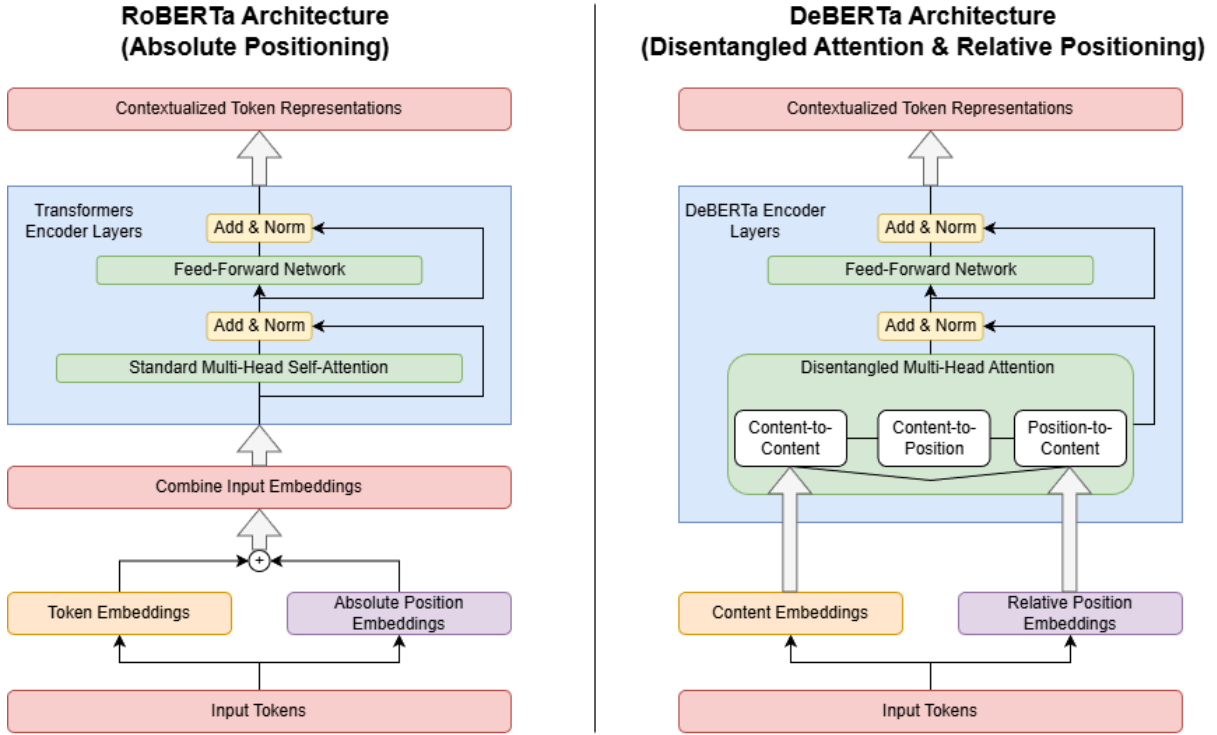


Figure 1: Structural comparison between RoBERTa and DeBERTa. Unlike RoBERTa, which sums absolute position and token embeddings at the input, DeBERTa processes content and relative positions independently through a disentangled attention mechanism.

2.2. Fine-tuning Approach

To prepare for the training process, we processed raw textual data collected from social media through a dedicated cleaning pipeline designed to remove noisy elements while preserving the contextual information necessary for effective modeling. Specifically, we normalized URL links and user mentions into placeholder tokens URL and @USER, respectively, while intentionally retaining hashtags in their original form to preserve salient topical cues. An empirical analysis of the sequence length distribution revealed that 95% of the samples contained fewer than 92 tokens; accordingly, we set the maximum sequence length (MAX_LENGTH) to 128 tokens. This choice ensures coverage of the vast majority of textual variations while maintaining computational efficiency.

To address the class imbalance (particularly in **run-2**), we implemented a custom error-penalization mechanism based on a weighted loss function. We computed class weights from the label frequencies using the `compute_class_weight` function from the `scikit-learn` library[15], thereby assigning greater importance to the minority class during optimization. In addition, we developed a custom **WeightedTrainer**, inheriting from the Hugging Face **Trainer**, to replace the default loss function with a Weighted Cross-Entropy Loss. This design explicitly encourages the model to learn more discriminative features for the minority class while reducing its tendency to bias predictions toward the. The loss function for a single sample is defined as:

$$\mathcal{L} = - \sum_{i=0}^{C-1} w_i \cdot y_i \log(\hat{y}_i) \quad (1)$$

Where:

- C : The total number of target classes, which is 2 for Task A (Reclaimed vs. Non-reclaimed).
- y_i : The ground truth binary indicator for class i .
- $\hat{y}_i$: The probability for class i predicted by the model (after the Softmax layer).

- w_i : The penalty weight assigned to class i to balance the gradient updates.

The class weights w_i were computed using the inverse frequency method via the `scikit-learn` library to ensure that the minority positive class receives higher importance during optimization:

$$w_i = \frac{N}{C \cdot n_i} \quad (2)$$

In this equation:

- N : Represents the total number of samples in the training set.
- n_i : Denotes the number of samples specifically belonging to class i .

We controlled the optimization process using a learning rate set to 2×10^{-5} , combined with a cosine decay learning rate schedule and a warm-up phase spanning the first 10% of the total training steps. Besides, we applied a weight decay[16] coefficient of 0.01 to enhance generalization performance and employed an early stopping mechanism with a patience of three epochs to prevent overfitting. We selected the final model checkpoint based on the highest macro-averaged F1 score on the validation set, as this metric represents the most appropriate and objective evaluation criterion for imbalanced classification tasks.

3. Data Augmentation Strategy

The empirical basis for this study consists of the datasets provided by the MultiPRIDE task organizers at EVALITA 2026. For Task A, which focuses on the classification of textual content, the provided material comprises social media data—specifically microblogging posts—annotated for the presence of reclamatory intent within the LGBTQ+ context.

While the primary data was originally distributed as language-specific subsets, a unified multilingual approach was adopted for this research. All individual language datasets were concatenated into a single corpus to facilitate the training of a robust multilingual model. This architectural decision was motivated by the objective of leveraging cross-lingual transfer learning, allowing the model to develop shared representations across different linguistic structures and cultural nuances associated with the target phenomenon.

3.1. MultiPride Dataset

The MultiPride dataset serves as the core corpus for the experimental framework, consisting of merged data from three languages: Italian, Spanish, and English. The dataset is characterized by a relatively balanced distribution among these languages, ensuring that the model is not unduly biased toward a single linguistic domain.

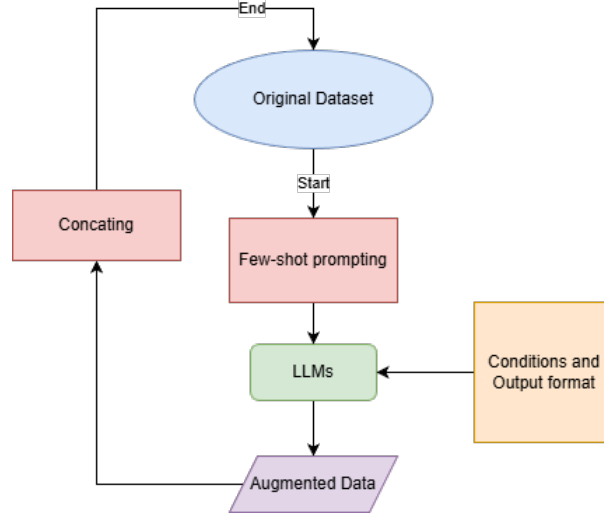
A significant feature of the dataset is the pronounced class imbalance[17] between the two target labels. While the majority of instances belong to the negative class (non-reclamation), the positive class (reclamation) is substantially underrepresented (see Table 1). Such an imbalance is representative of real-world data distributions in social media monitoring, where specific linguistic phenomena like reclamation occur with lower frequency than general discourse. This disparity presents a notable challenge for supervised learning, as models may develop a bias toward the majority class without specific intervention.

3.2. Synthetic Data

To address the limitations inherent in the original dataset distribution, particularly the sparsity of the minority class, a synthetic data generation phase was implemented. Large Language Models (LLMs)[18] were utilized to expand the training set through a few-shot prompting strategy[19]. We provided the models with representative examples from the original MultiPride corpus, and the generation process

Table 1*Distribution of samples in the MultiPride dataset*

Split	Non-reclaimed	Reclaimed	Total
Training	2,048	342	2,390
Validation	512	86	598
Total	2,560	428	2,988

**Figure 2:** Workflow of the iterative data augmentation process using Few-shot prompting and Large Language Models (LLMs)**Table 2***Distribution of samples in the MultiPRIDE dataset after data augmentation.*

Split	Non-reclaimed	Reclaimed	Total
Training (Augmented)	2,505	2,440	4,945
Validation	512	86	598
Total	3,017	2,526	5,543

ensured that the synthetic outputs maintained semantic and stylistic consistency with the source data - the workflow is visualized in Figure 2.

Synthetic augmentation primarily targeted the underrepresented label 1 to mitigate class imbalance and prevent majority-class bias. Furthermore, this strategy served to scale the training corpus for more stable weight updates during fine-tuning while simultaneously diversifying linguistic contexts to enhance model generalization on unseen instances. The use of advanced LLMs enabled the creation of high-quality synthetic examples that accurately reflect the complex pragmatic nature of reclaimed language, while preserving the overall distribution of the task’s domain[20].

Following this workflow, we applied the augmentation process to the MultiPRIDE dataset. First, we performed data augmentation for each target language by prompting Large Language Models (LLMs) to generate synthetic samples in .csv format. These generated files were then merged with our existing training set, while the validation set remained untouched to ensure the integrity of the evaluation. The final composition of the augmented dataset is presented in detail in Table 2:

Table 3*Classification results across different languages and runs*

Language	Run	F1 (Class 0)	F1 (Class 1)	Macro F1
English (en)	run-1	0.9396	0.3214	0.6305
	run-2	0.9402	0.2400	0.5901
Spanish (es)	run-1	0.9312	0.5868	0.7590
	run-2	0.9365	0.6049	0.7707
Italian (it)	run-1	0.9396	0.7418	0.8407
	run-2	0.9475	0.7704	0.8589

Table 4*Our results in MultiPRIDE Task A at EVALITA 2026 Final Ranking*

Subtask	Team	Run	Rank	Macro F1
A1 (Italian)	Ghavidel-Rajabi	1	1	0.8981
	MilaNLP	1	2	0.8959
	Baseline	1	10	0.8731
	NamDang (Ours)	2	12	0.8589
	NamDang (Ours)	1	17	0.8407
A2 (Spanish)	The Hate Busters	2	1	0.7776
	NamDang (Ours)	2	2	0.7707
	NamDang (Ours)	1	3	0.7590
	Baseline	1	14	0.7000
A3 (Italian)	Avahi	1	1	0.6416
	SaFe Tweets	1	2	0.6329
	NamDang (Ours)	1	3	0.6305
	NamDang (Ours)	2	7	0.5901
	Baseline	1	8	0.5760

4. Result

The system performance was evaluated through two official runs on the MultiPRIDE test set at EVALITA 2026. The objective of these experiments was to compare the generalization capabilities of different Transformer architectures under the influence of data augmentation strategies versus training on the original dataset. **run-1** corresponds to a configuration using the microsoft/mdeberta-v3-base model, which incorporates architectural enhancements and is trained on the augmented dataset. In contrast, **run-2** serves as a reference baseline, employing the multilingual xlm-roberta-base model trained directly on the original dataset.

The experimental results are summarized using a macro-averaged F1 score, with the latter serving as the primary evaluation metric for assessing classification performance on imbalanced datasets. Table 3 reports the detailed system performance across the three target languages (English, Spanish, and Italian) for both runs. Overall, the system achieved its best performance on Italian, with the highest macro-averaged F1 score reaching 0.8589 in **run-2**. Spanish ranked second, attaining a peak macro F1 score of 0.7707, also in **run-2**. In contrast, English proved to be the most challenging language for the system, with macro F1 scores ranging only from 0.5901 to 0.6305 across the two runs. This performance gap may reflect differences in contextual complexity or language-specific patterns in how reclaimed language is used by LGBTQ+ communities on social media across different countries.

For English, **run-1** yielded comparatively better results for the positive class and in terms of macro F1, suggesting that the combination of data augmentation and the disentangled attention mechanism of DeBERTa facilitated the learning of more diverse lexical variants. Conversely, for Spanish and

Italian, **run-2** consistently outperformed **run-1** across all evaluation metrics. This observation may indicate that, for these two languages, the original data distribution was sufficiently informative for XLM-RoBERTa to exploit effectively, or that the data augmentation applied in **run-1** may have inadvertently introduced noise, leading to a degradation in local classification accuracy.

5. Discussion

Following the official evaluation phase, the organizing committee released detailed results for each submitted run, along with an `errors_lang.csv` file for every language and subtask, containing 20 representative misclassified instances. These error files correspond to the best-performing run according to the overall results. An in-depth inspection of these instances provides valuable insight into the underlying causes of the model’s incorrect predictions. Based on the analysis of the error files across the three languages, we identify several recurring patterns that highlight the system’s limitations.

For Italian and Spanish, the model errors primarily stem from the use of slur terms within socially and politically complex contexts, where the boundary between attack and reclamation becomes blurred. In Italian, the system struggles in particular with meta-discursive utterances, in which users quote or explicitly discuss the use of slurs rather than directly targeting an individual or group, leading to misinterpretation of the communicative intent. Moreover, highly polarized Pride-related contexts, characterized by a dense concentration of sensitive terms employed with positive or empowering meanings, tend to trigger default negative predictions due to the model’s over-reliance on surface-level lexical cues. Similarly, in counter-speech against discrimination, users often deliberately reuse slur terms to criticize oppressive behaviors; however, the model lacks the pragmatic depth required to distinguish social critique from direct verbal abuse. In Spanish, the presence of hashtags such as `#OrgulloLGTBI` or `#Pride` introduces additional noise, occasionally biasing the model toward positive labels even when the surrounding content still contains strong or potentially offensive language. Furthermore, regional variants and colloquial nuances of certain slurs are not sufficiently captured by the model, particularly under the non-augmented training setting, resulting in systematic classification errors.

English is the language in which the model exhibits the weakest performance, largely due to the widespread and culturally entrenched nature of reclaimed language, which creates extremely fragile boundaries between offense and empowerment. In many instances, distinguishing between in-group and out-group usage depends almost entirely on the speaker’s identity—an aspect that cannot be inferred by a text-only model. Additionally, forms of dark humor, sarcasm, and self-directed irony are pervasive in the data, leading the system to assign negative labels based solely on the presence of slur tokens, while ignoring narrative structure and speaker stance. The polysemy of terms such as “queer”, which may function as a positive academic or political identifier or as an insult in conservative contexts, further exacerbates prediction instability and bias.

6. Conclusion

This paper reports our participation in Task A of the MultiPride shared task at EVALITA 2026, focusing on the identification of reclaimed language within the LGBTQ+ community across English, Italian, and Spanish. The system achieved its best performance (F1-macro) in Italian (0.8589) and Spanish (0.7707) using a fine-tuned XLM-RoBERTa-base model trained on the original dataset, while for English the highest score (0.6305) was obtained with mDeBERTa-v3-base enhanced by synthetic data augmentation, highlighting the benefits of disentangled attention for capturing lexical diversity.

As future work, we aim to incorporate identity-aware modeling to better distinguish in-group from out-group usage, a key limitation of text-only approaches. We also plan to explore contrastive learning and external knowledge integration to address pragmatic phenomena such as sarcasm, self-directed irony, and the polysemy of sensitive terms, as well as to enrich synthetic data with more culturally and pragmatically complex contexts.

Declaration on Generative AI

In this work, Generative AI tools were used exclusively for linguistic clarity. The conceptualization, data analysis, and conclusions remain the sole work of the authors, who accept full responsibility for the manuscript's content.

References

- [1] M. S. Jahan, M. Oussalah, A systematic review of hate speech automatic detection using natural language processing, *Neurocomput.* 546 (2023). URL: <https://doi.org/10.1016/j.neucom.2023.126232>. doi:10.1016/j.neucom.2023.126232.
- [2] A. Velankar, H. Patil, R. Joshi, A review of challenges in machine learning based automated hate speech detection, 2022. URL: <https://arxiv.org/abs/2209.05294>. arXiv:2209.05294.
- [3] J. G. Parmenter, R. V. Galliher, A. D. A. Maughan, An exploration of lgbtq+ community members' positive perceptions of lgbtq+ culture, *The Counseling Psychologist* 48 (2020) 1016–1047. URL: <https://doi.org/10.1177/0011000020933188>. doi:10.1177/0011000020933188. arXiv:<https://doi.org/10.1177/0011000020933188>.
- [4] A. D. Galinsky, C. S. Wang, J. A. Whitson, E. M. Anicich, K. Hugenberg, G. V. Bodenhausen, The reappropriation of stigmatizing labels: The reciprocal relationship between power and self-labeling, *Psychological Science* 24 (2013) 2020–2029. URL: <https://doi.org/10.1177/0956797613482943>. doi:10.1177/0956797613482943. arXiv:<https://doi.org/10.1177/0956797613482943>, PMID: 23955354.
- [5] M. Melis, G. Lapesa, D. Assenmacher, A modular taxonomy for hate speech definitions and its impact on zero-shot LLM classification performance, in: A. Calabrese, C. de Kock, D. Nozza, F. M. Plaza-del Arco, Z. Talat, F. Vargas (Eds.), *Proceedings of the The 9th Workshop on Online Abuse and Harms (WOAH)*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 490–521. URL: <https://aclanthology.org/2025.woah-1.45/>.
- [6] M. Tonneau, D. Liu, N. Malhotra, S. A. Hale, S. P. Fraiberger, V. Orozco-Olvera, P. Röttger, Hateday: Insights from a global hate speech dataset representative of a day on twitter, 2025. URL: <https://arxiv.org/abs/2411.15462>. arXiv:2411.15462.
- [7] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: *Neural Information Processing Systems*, 2017. URL: <https://api.semanticscholar.org/CorpusID:13756489>.
- [9] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020. URL: <https://arxiv.org/abs/1911.02116>. arXiv:1911.02116.
- [10] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, 2023. URL: <https://arxiv.org/abs/2111.09543>. arXiv:2111.09543.
- [11] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, E. Hovy, A survey of data augmentation approaches for nlp, 2021. URL: <https://arxiv.org/abs/2105.03075>. arXiv:2105.03075.
- [12] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. URL: <https://arxiv.org/abs/1910.03771>. arXiv:1910.03771.
- [13] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov,

- Roberta: A robustly optimized bert pretraining approach, 2019. URL: <https://arxiv.org/abs/1907.11692>. arXiv:1907.11692.
- [14] P. He, X. Liu, J. Gao, W. Chen, Deberta: Decoding-enhanced bert with disentangled attention, 2021. URL: <https://arxiv.org/abs/2006.03654>. arXiv:2006.03654.
- [15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, Édouard Duchesnay, Scikit-learn: Machine learning in python, 2018. URL: <https://arxiv.org/abs/1201.0490>. arXiv:1201.0490.
- [16] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, 2019. URL: <https://arxiv.org/abs/1711.05101>. arXiv:1711.05101.
- [17] S. Henning, W. Beluch, A. Fraser, A. Friedrich, A survey of methods for addressing class imbalance in deep-learning based natural language processing, in: A. Vlachos, I. Augenstein (Eds.), Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Dubrovnik, Croatia, 2023, pp. 523–540. URL: <https://aclanthology.org/2023.eacl-main.38/>. doi:10.18653/v1/2023.eacl-main.38.
- [18] A. Matarazzo, R. Torlone, A survey on large language models with some insights on their capabilities and limitations, 2025. URL: <https://arxiv.org/abs/2501.04040>. arXiv:2501.04040.
- [19] T. Schick, H. Schütze, True few-shot learning with prompts—a real-world perspective, Transactions of the Association for Computational Linguistics 10 (2022) 716–731. URL: https://doi.org/10.1162/tac1_a_00485. doi:10.1162/tac1_a_00485.
- [20] B. Ding, C. Qin, R. Zhao, T. Luo, X. Li, G. Chen, W. Xia, J. Hu, A. T. Luu, S. Joty, Data augmentation using large language models: Data perspectives, learning paradigms and challenges, 2024. URL: <https://arxiv.org/abs/2403.02990>. arXiv:2403.02990.

NetGuardAI at MultiPRIDE: Multilingual Detection of Reclaimed Language

Noe Come Jacques LE POLLTEC^{1,†}, Elena Simona APOSTOL^{1,†} and
Ciprian Octavian TRUICĂ^{1,2,*,†}

¹Computer Science and Engineering Department, Faculty of Automatic Control and Computers, National University of Science and Technology Politehnica Bucharest, Splaiul Independenței 313, Bucharest, 060042, Romania

²Academy of Romanian Scientists, Ilfov 3, Bucharest, 050044, Romania

Abstract

Online social media platforms have been widely recognized as environments for the rapid propagation of hate speech targeting specific communities, which poses significant challenges for NLP systems. In particular, the phenomenon of semantic reclamation complicates automatic moderation, as slurs may be used either offensively or as expressions of identity and solidarity within marginalized groups. This paper describes the participation of the NetGuardAI team in the MultiPRIDE shared task at EVALITA 2026. The task focuses on distinguishing between the hateful use of slurs and their “reclaimed” use by the LGBTQ+ community in Italian, English, and Spanish. The core challenge is the significant class imbalance and the linguistic nuance required to separate hate speech from self-identification. Our approach utilizes a multilingual DistilBERT model fine-tuned with a weighted Cross-Entropy Loss. We participated in both Task A (Text-only) and Task B (Text + Biography). Our best performance was achieved in Italian Task A, reaching a Macro F1-score of 0.854. The results highlight the effectiveness of weighted loss for Italian, but also reveal challenges in cross-lingual generalization and low recall for minority classes in English and Spanish.

Warning: This paper contains examples of explicitly offensive content.

Keywords

Hate Speech Detection, Reclaimed Language, DistilBERT, Multilingual NLP, Class Imbalance, EVALITA 2026

1. Introduction

Online hate speech detection often suffers from a critical bias, i.e., the inability to distinguish between a slur used to attack and a slur used to self-identify (reclamation). This issue is particularly acute for the LGBTQ+ community. According to the 2024 European Union Agency for Fundamental Rights (FRA) survey [1], over 50% of LGBTQ+ respondents experienced hate-motivated harassment, a significant increase from 2019. Similarly, reports from ILGA-Europe highlight that online platforms are increasingly becoming vectors for normalized hate speech and disinformation campaigns.

The MultiPRIDE [2] shared task at EVALITA 2026 [3] addresses this problem by focusing on the Multilingual Automatic Detection of Reclamation of Slurs. The goal is to identify whether a slur is used aggressively or as an act of “reappropriation” and solidarity within the community. The challenge is divided into two subtasks:

- **Task A (Textual Content):** Participants must classify the tweet as “Reclaimed” (Label 1) or “Not Reclaimed/Hate” (Label 0) using only the provided textual content.
- **Task B (Contextual Content):** Participants may utilize the user’s profile biography (when available) in addition to the tweet text to aid classification.

The submitted solutions are evaluated using the Macro F1-score computed over the binary labels.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ noe_come.le@stud.acs.upb.ro (N. C. J. LE POLLTEC); elena.apostol@upb.ro (E. S. APOSTOL); ciprian.truica@upb.ro (C. O. TRUICĂ)

ORCID: 0009-0009-4682-6629 (N. C. J. LE POLLTEC); 0000-0001-6397-4951 (E. S. APOSTOL); 0000-0001-7292-4462 (C. O. TRUICĂ)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The core challenge of the MultiPRIDE dataset provided for this competition is the linguistic nuance required to separate these classes, combined with a significant class imbalance where reclaimed examples are the minority. Standard models often fail to capture the context required to understand if a slur is being used affectionately by an in-group member or maliciously by an outsider.

In this paper, we present NetGuardAI, a Transformers-based solution to tackle the online hate speech problem in the LGBTQ+ community. We address the task objectives by deploying a lightweight transformer architecture (i.e., DistilBERT) optimized for imbalanced learning via cost-sensitive training. We evaluate the impact of using only textual content (Task A) versus enriching the input with user biographies (Task B).

The remainder of this paper is organized as follows: Section 2 reviews the state of the art in hate speech detection. Section 3 details the methodology, including the DistilBERT architecture and weighted loss strategy. Section 4 presents the exploratory data analysis, while Section 5 provides experimental results and error analysis. Section 6 provides a critical discussion of our findings. Finally, Section 7 concludes the paper and outlines future work.

2. Related work

The task of detecting hate speech has evolved significantly in recent years. Early approaches relied on lexicon-based methods and N-gram models. However, Davidson et al. [4] demonstrated that these methods struggle with the contextual dependence of abusive language, often leading to high false-positive rates when slurs are present but used non-offensively.

The advent of Transformers, introduced by Devlin et al. [5] with BERT (Bidirectional Encoder Representations from Transformers), revolutionized the field. BERT utilizes a self-attention mechanism to process words in relation to all other words in a sentence, capturing long-distance dependencies crucial for understanding intent.

The current literature discusses a wide range of techniques and models for harmful content detection, together with novel approaches aimed at mitigating its online diffusion. For harmful content detection, methods employ different techniques, such as word embeddings [6], document-level representations [7], transformer-based architectures [8, 9, 10, 11], ensembles or mixtures of transformers [12, 13, 14], as well as large language models [15]. Some approaches that incorporate multimodal data representations that leverage social context [16] and information propagation [17] manage to further improve detection effectiveness. Research on network immunization introduces innovative strategies to limit the spread of harmful content across social media platforms [18, 19, 20, 21, 22]. Furthermore, real-time systems for identifying and counteracting the online diffusion of harmful content have been developed to protect users and communities [23, 24, 25, 26, 27].

For the MultiPRIDE task [3], which involves multiple languages (i.e., Italian, English, Spanish), multilingual models are essential. While mBERT and XLM-RoBERTa are standard baselines, they are computationally expensive. Sanh et al. [28] proposed DistilBERT, a distilled version of BERT. According to their research, DistilBERT retains 97% of BERT’s performance on downstream tasks while being 40% smaller and 60% faster. This efficiency makes it an ideal candidate for constrained resource environments while maintaining high accuracy in text classification.

3. Methodology

Our system employs a supervised learning approach, utilizing fine-tuning of pre-trained Transformers. The overall pipeline is illustrated in Figure 1.

3.1. Architecture

The core components of our architecture are detailed below.

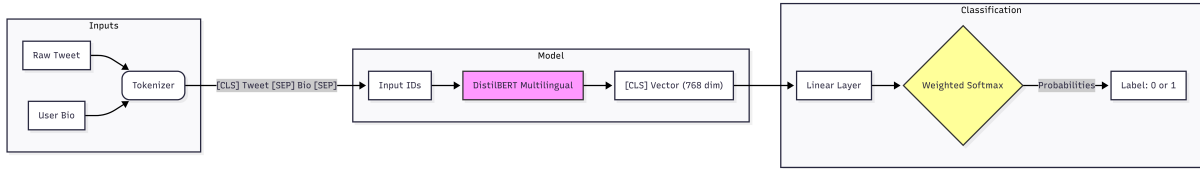


Figure 1: The NetGuardAI Pipeline: (1) Raw text and Bio are tokenized, (2) Input IDs are fed to DistilBERT, (3) A classification head predicts the label using Weighted Loss.

1. **Input Construction & Tokenization:** The system accepts two inputs: the raw tweet text and, for Task B, the user biography. These are concatenated using the special separator token [SEP]. The sequence is then processed by the *DistilBERT Tokenizer*, which splits words into sub-word units (WordPiece algorithm). This handles out-of-vocabulary terms effectively, which is critical for the slang-heavy language found in LGBTQ+ discourse. The sequence is prepended with the [CLS] classification token and truncated/padded to a fixed length of 128 tokens.
2. **DistilBERT Backbone:** This component serves as the semantic encoder of our architecture. The token IDs are fed into the `distilbert-base-multilingual-cased` model. This model consists of 6 Transformer encoder layers (half the depth of BERT) and approximately 134 million parameters. It applies self-attention mechanisms to generate context-aware embeddings for every token, capturing the semantic relationship between the slur and the surrounding words (e.g., distinguishing "I am [slur]" from "You are [slur]").
3. **Contextual Pooling:** From the output of the final transformer layer, we extract the vector corresponding to the [CLS] token. This 768-dimensional vector serves as the aggregate representation of the entire sequence’s semantic meaning, acting as a summary of the input text/bio combination.
4. **Classification Head & Weighted Loss:** The pooled [CLS] vector is passed through a fully connected linear layer (Dropout $p = 0.1$) that projects the 768 dimensions down to 2 logits (Reclaimed vs. Hate). During training, we apply a **Weighted Cross-Entropy Loss**. This loss function assigns a higher scalar weight to the minority class (Class 1), penalizing the model more severely for misclassifying reclaimed tweets compared to hate tweets, thereby counteracting the 11:1 data imbalance.

3.2. Input Representation

We define the input strategy distinctively for the two subtasks:

- **Task A (Textual Content):** The raw tweet text was tokenized directly using the DistilBERT tokenizer.
- **Task B (Contextual Content):** For Italian and Spanish, we concatenated the tweet text and the user biography using the separator token: [CLS] Tweet [SEP] Bio [SEP].

3.3. Training Strategy

Given the dominance of the “Not Reclaimed” class (0) (approx. 11:1 ratio), we implemented a **Weighted Cross-Entropy Loss**. We calculated class weights inversely proportional to class frequencies in the training set. The models were trained for 2 epochs using the AdamW optimizer.

4. Exploratory Data Analysis

The MultiPRIDE dataset consists of tweets in three languages, i.e., Italian, English, and Spanish. We performed statistical analysis and NMF (Non-negative Matrix Factorization) topic modeling [29] to understand the data characteristics.

4.1. Text Statistics

We analyzed the word count distribution across languages and classes (Figure 2).

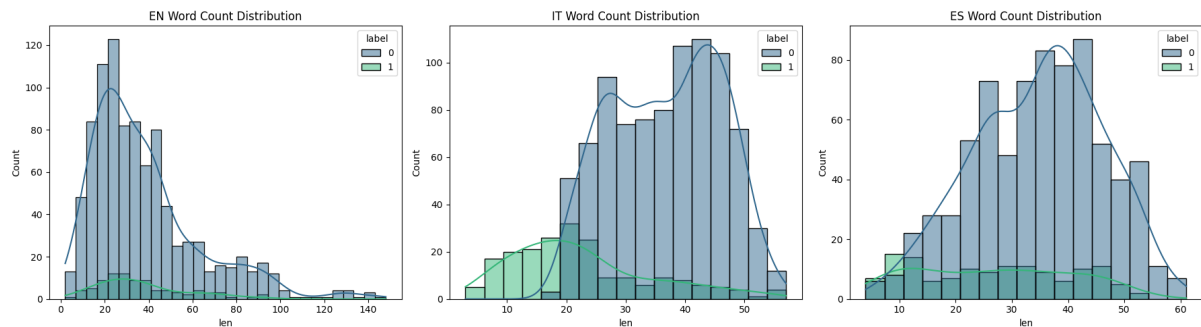


Figure 2: Distribution of tweet lengths per language. Italian Reclaimed tweets (Label 1) are notably shorter and more uniform than English ones.

We observed a significant structural difference in Italian: Reclaimed tweets are significantly shorter (mean 22.5 words) compared to Hate tweets (36.8 words). This suggests that Italian reclamation often appears in short slogans or self-declarations, making it distinct from the longer, more complex hate speech. In contrast, the English and Spanish distributions are more overlapping.

4.2. Topic Modeling

We applied NMF Topic Modeling (N=4 components) to the "Reclaimed" class. The results are presented in Table 1.

Table 1

Top Keywords per Topic (NMF, 4 Components) for Reclaimed Tweets

Lang	Topic	Keywords
EN	1	like, use, word, don, really
EN	2	gay, faggot, time, friends, oh
EN	3	tranny, used, slur, people, drag
EN	4	queer, reclaimed, faggot, way, day
IT	1	di, che, la, per, in
IT	2	user, frocia, perché, lgbt, mia
IT	3	forci, url, sono, di, tutti
IT	4	sono, mi, non, io , lella
ES	1	que, la, de, el, no
ES	2	orgullo, feliz, orgullolgtbi, pride, maricones
ES	3	user, de, maricón, url, orgullo2021
ES	4	las, por, de, en, el

The topic analysis reveals why detection varies by language:

- **English (Meta-Linguistic Ambiguity):** Topics 1 and 3 are dominated by discussion verbs and nouns ("use", "word", "used", "slur") rather than identity markers. This indicates that English users often discuss the *nature* of the words themselves, creating a high level of ambiguity between educational/meta-discussion and actual reclamation.
- **Italian (Self-ID):** Topic 4 contains explicit first-person pronouns ("sono", "mi", "io" → "I am", "me", "I"). This clear pattern of self-identification serves as a strong feature for the model, contributing to the high F1 score (0.85).

- **Spanish (Event-Based):** Topic 2 is dominated by "Pride" event keywords ("orgullo", "feliz"). While distinct, the remaining topics are noisy with stopwords ("que", "de", "el"), suggesting that reclamation in Spanish is more context-dependent than in Italian.

5. Experimental results

We evaluated our models on the official MultiPRIDE test sets. The official metric for the task is the Macro F1-score. Table 2 summarizes our performance.

Table 2

Official Results (Macro F1) for NetGuardAI on the Test Set.

Language	Task	Macro F1	F1 (Class 0)	F1 (Class 1)
Italian	A	0.854	0.942	0.765
English	A	0.548	0.908	0.187
Spanish	A	0.648	0.833	0.462
Italian	B	0.740	0.880	0.601
Spanish	B	0.493	0.920	0.065

5.1. Setup Documentation

The experiments were conducted on the Google Colab platform using the following environment:

- **Hardware:** NVIDIA Tesla T4 GPU.
- **Libraries:** Transformers 4.57.3, PyTorch 2.9.0, Scikit-learn 1.6.1.

The code is publicly available on GitHub at https://github.com/DS4AI-UPB/NetGuardAI_MultiPride2026.

6. Limitations and Discussion

6.1. Performance Analysis

Our results show a distinct pattern: Class 0 (Hate) consistently achieves higher F1-scores than Class 1 (Reclaimed). This is correlated with the class imbalance (11:1). However, we observe that Italian outperforms English and Spanish ($F1_{macro} = 0.854$ vs 0.548, see Table 2).

Based on our EDA (Table 1), we attribute the high performance in Italian to the prevalence of **explicit self-identification markers** (Topic 4: "sono", "io"). The model easily learns that "I am [slur]" corresponds to Reclamation.

6.2. Error Analysis

We analyzed specific misclassifications from the error logs to understand model failure points.

Italian (Contextual Complexity):

- *Example (ID it_686):* "Sarò anche molto frocio, ma...". The model predicted Hate (0) likely due to aggressive sexual language later in the tweet, overlooking the initial self-identification marker ("Sarò anche" - "I may be").
- *Example (ID it_1767):* Tweets reporting on homophobia (e.g., discussing people shouting "Finocchio" in a piazza) were sometimes misclassified as Reclaimed (1) due to the presence of LGBTQ+ keywords without direct negative sentiment toward the user.

English (Ambiguity):

- *Example (ID en_707):* "throw another faggot on the fire". The model failed to distinguish the archaic/literal meaning (bundle of sticks) from the slur.
- *Example (ID en_396):* "I'm a degenerate piece of faggot trash". The model predicted Hate (0) due to strong negative keywords ("degenerate", "trash"), missing the self-deprecating humor typical of reclamation.

Spanish (Noise in Task B):

In Task B, performance dropped ($F1 = 0.065$ for Class 1). The addition of User Bios diluted the signal. For example, many bios contained generic text ("Music lover", "Madrid") which acted as noise (stopwords seen in Table 1, Topic 1) rather than the "Pride" markers found in Topic 2.

7. Conclusions

In this paper, we presented the NetGuardAI submission for the MultiPRIDE 2026 shared task. We developed a multilingual hate speech detection system based on a fine-tuned DistilBERT architecture, employing a weighted cross-entropy loss to address the severe class imbalance inherent in reclamation detection.

Our experimental results demonstrate a significant disparity in performance across languages. The system achieved its best results in Italian ($F1_{macro} = 0.854$), successfully leveraging explicit self-identification markers (e.g., "sono", "io") to distinguish reclaimed usage. However, the model struggled with the semantic ambiguity of English ($F1_{macro} = 0.548$) and Spanish ($F1_{macro} = 0.648$), where reclamation often involves complex meta-linguistic discussions or context-dependent irony that a lightweight model fails to capture with limited training data. Furthermore, our analysis of Task B reveals that simply concatenating user biographies can introduce noise, as evidenced by the performance drop in Spanish, suggesting that irrelevant bio information dilutes the textual signal.

For future work, we plan to investigate three main directions. First, we aim to implement selective attention mechanisms that can dynamically weight the importance of user biographies, filtering out generic information while attending to relevant identity markers. Second, to further mitigate class imbalance, we propose exploring data augmentation techniques, such as back-translation or generative synthetic data, to increase the diversity of the minority "reclaimed" class. Finally, we intend to evaluate larger multilingual models (e.g., XLM-RoBERTa) to determine whether increased model capacity can better resolve the linguistic ambiguities observed in English.

Acknowledgments

We thank the organizers of EVALITA 2026 for providing the dataset and the shared task platform.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini to:

- Write code for the DistilBERT training loop and data processing.
- Summarize patterns in the error analysis logs.
- Refine the English grammar and structure of the LaTeX document.

After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] E. U. A. for Fundamental Rights, Fundamental rights report 2024, 2024. URL: <https://fra.europa.eu/en/publication/2024/fundamental-rights-report-2024>, last accessed: 2026-01-12.

- [2] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [4] T. Davidson, D. Warmesley, M. Macy, I. Weber, Automated hate speech detection and the problem of offensive language, in: Proceedings of the 11th International AAAI Conference on Web and Social Media, 2017, pp. 512–515.
- [5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT, 2019.
- [6] V.-I. Ilie, C.-O. Truică, E.-S. Apostol, A. Paschke, Context-Aware Misinformation Detection: A Benchmark of Deep Learning Architectures Using Word Embeddings, IEEE Access 9 (2021) 162122–162146. doi:10.1109/ACCESS.2021.3132502.
- [7] C.-O. Truică, E.-S. Apostol, It's all in the Embedding! Fake News Detection using Document Embeddings, Mathematics 11 (2023) 1–29(508). doi:10.3390/math11030508.
- [8] C.-O. Truică, E.-S. Apostol, MisRoBERTa: Transformers versus Misinformation, Mathematics 10 (2022) 1–25(569). doi:10.3390/math10040569.
- [9] C.-O. Truică, E.-S. Apostol, A. Paschke, Awakened at CheckThat! 2022: Fake News Detection using BiLSTM and sentence transformer, in: Working Notes of the Conference and Labs of the Evaluation Forum, 2022, pp. 749–757.
- [10] M.-D. Cotelin, E.-S. Apostol, C.-O. Truică, NetGuardAI at EXIST2025: Sexism Detection using mDeBERTa, in: Working Notes of the Conference and Labs of the Evaluation Forum, 2025, pp. 1897–1905.
- [11] D.-E. Burghilea, C.-O. Truică, E.-S. Apostol, Verit-albert: A finetuned llm approach for verifying information credibility, in: The 2025 24th RoEduNet Conference: Networking in Education and Research (RoEduNet), 2025, pp. 1–6. doi:10.1109/RoEduNet68395.2025.11208267.
- [12] A. Petrescu, C.-O. Truică, E.-S. Apostol, Language-based Mixture of Transformers for EXIST2024, in: Working Notes of the Conference and Labs of the Evaluation Forum, 2024, pp. 1157–1164.
- [13] A. Petrescu, C.-O. Truică, E.-S. Apostol, Language-based Mixture of Transformers for Sexism Identification in Social Networks, in: Conference and Labs of the Evaluation Forum, 2025, pp. 142–155. doi:10.1007/978-3-032-04354-2_10.
- [14] A. Petrescu, E.-S. Apostol, C.-O. Truică, Awakened at EXIST2025: Adaptive Mixture of Transformers, in: Working Notes of the Conference and Labs of the Evaluation Forum, 2025, pp. 2112–2118.
- [15] Ciprian-Octavian, E.-S. Apostol, A.-G. Ilie, A. Paschke, HarmLLaMA: Harmful Language Detection with Large Language Models, in: International Conference on Intelligent Computer Communication and Processing (ICCP 2025), 2025.
- [16] C.-O. Truică, E.-S. Apostol, P. Karras, DANES: Deep Neural Network Ensemble Architecture for Social and Textual Context-aware Fake News Detection, Knowledge-Based Systems 294 (2024) 1–13(111715). doi:https://doi.org/10.1016/j.knosys.2024.111715.
- [17] C.-O. Truică, E.-S. Apostol, M. Marogel, A. Paschke, GETAE: Graph Information Enhanced Deep Neural Network Ensemble Architecture for fake news detection, Expert Systems with Applications 275 (2025) 126984. doi:10.1016/j.eswa.2025.126984.
- [18] C.-O. Truică, E.-S. Apostol, T. Ștefu, P. Karras, A Deep Learning Architecture for Audience Interest Prediction of News Topic on Social Media, in: International Conference on Extending Database Technology (EDBT2021), 2021, pp. 588–599. doi:10.5441/002/EDBT.2021.69.
- [19] A. Petrescu, C.-O. Truică, E.-S. Apostol, P. Karras, Sparse shield: Social network immunization vs. harmful speech, in: ACM International Conference on Information & Knowledge Management,

- 2021, pp. 1426–1436. doi:10.1145/3459637.3482481.
- [20] C.-O. Truică, E.-S. Apostol, R.-C. Nicolescu, P. Karras, MCWDST: A Minimum-Cost Weighted Directed Spanning Tree Algorithm for Real-Time Fake News Mitigation in Social Media, *IEEE Access* 11 (2023) 125861–125873. doi:10.1109/ACCESS.2023.3331220.
 - [21] E.-S. Apostol, Özgür Coban, C.-O. Truică, CONTAIN: A community-based algorithm for network immunization, *Engineering Science and Technology, an International Journal* 55 (2024) 1–10(101728). doi:<https://doi.org/10.1016/j.jestch.2024.101728>.
 - [22] A. Petrescu, C.-O. Truică, E.-S. Apostol, A. Paschke, EDSA-Ensemble: An Event Detection Sentiment Analysis Ensemble Architecture, *IEEE Transactions on Affective Computing* 16 (2025) 555–572. doi:10.1109/taffc.2024.3434355.
 - [23] E.-S. Apostol, C.-O. Truică, A. Paschke, ContCommRTD: A Distributed Content-Based Misinformation-Aware Community Detection System for Real-Time Disaster Reporting, *IEEE Transactions on Knowledge and Data Engineering* (2024) 1–12. doi:10.1109/tkde.2024.3417232.
 - [24] C.-O. Truică, A.-T. Constantinescu, E.-S. Apostol, StopHC: A Harmful Content Detection and Mitigation Architecture for Social Media Platforms, in: *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP 2024)*, 2024, pp. 1–5. doi:10.1109/ICCP63557.2024.10793051.
 - [25] M.-D. Cotelin, C.-O. Truică, E.-S. Apostol, Cleannews: a network-aware fake news mitigation architecture for social media, *arXiv preprint arXiv:2509.04489* (2025).
 - [26] C.-O. Truică, J. Darmont, J. Velcin, A scalable document-based architecture for text analysis, in: *International Conference on Advanced Data Mining and Applications*, Springer, 2016, pp. 481–494. doi:10.1007/978-3-319-49586-6_33.
 - [27] C.-O. Truică, A. Guille, M. Gauthier, Cats: Collection and analysis of tweets made simple, in: *ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*, ACM, 2016, pp. 41–44. doi:10.1145/2818052.2874320.
 - [28] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, *arXiv preprint arXiv:1910.01108* (2019).
 - [29] C.-O. Truică, E. S. Apostol, C. A. Leordeanu, Topic modeling using contextual cues, in: *IEEE International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC2017)*, 2017, pp. 203–210. doi:10.1109/synasc.2017.00041.

SaFe Tweets at MultiPRIDE: From Multi-Model Ensembles to Expert LLMs for Reclamation Detection

Sara Visconti[†], Federica Manzi^{1,†}

¹*Independent Researcher*

¹*Università degli Studi di Torino - Dipartimento di Informatica, Corso Svizzera 185, 10149 Torino, Italy*

Abstract

In this work, we describe our participation in the EVALITA 2026 MultiPRIDE task, which focuses on classifying Italian and English tweets by distinguishing between reclaiming and non-reclaiming uses of potentially offensive terms against the LGBTQIA+ community. In particular, we participated in subtasks A1 (Italian), A3 (English), and B1 (Italian with author context). For Italian in both subtasks, we employed an ensemble of lightweight models using majority voting to combine their predictions. The lightweight models included a logistic regression classifier, SetFit [1], and XLM-RoBERTa [2]. For English, due to the lack of such linguistic cues and the more severe class imbalance, we adopted a more complex two-step pipeline: a DeBERTaV3 [3] model with LoRA [4] fine-tuning for initial filtering, followed by LLaMa 4 Scout [5] to refine low-confidence predictions. Despite leveraging powerful models, classification remained challenging due to annotation subjectivity, ambiguous tweets, and limited contextual information. Our results highlight the importance of both model choice and contextual understanding in distinguishing reclaiming from non-reclaiming language.

Keywords

reclaimed language, hate-speech detection, LGBTQIA+ tweets, social media analysis, context-aware classification, EVALITA 2026 MultiPRIDE, ensemble models, transformer-based models, LLMs pipeline, LoRA fine-tuning

Warning: this paper contains examples of content that, even though obfuscated, some readers may find offensive.

1. Introduction

Contextual information is crucial in the automatic detection of hate speech and offensive language, especially for reclaimed language, where terms historically used as slurs are reclaimed and used in positive or neutral ways by members of the targeted community. This distinction is important because, without it, social media platforms relying on automated detection systems might incorrectly flag reclaimed content as toxic, effectively censoring members of the LGBTQIA+ community [6]. For this reason, it is essential to develop systems that allow for a more nuanced and context-dependent distinction between such cases to avoid mislabeling reclaimed language, while still identifying genuinely offensive content [7].

In this paper, we describe our participation in the MultiPRIDE shared task at EVALITA 2026 [8]. The task is formulated as a binary classification problem, where systems are required to determine whether words and expressions in tweets that are potentially offensive toward the LGBTQIA+ community are used in a reclaiming or non-reclaiming way by the author of the tweet. The task is divided into two main subtasks:

- **Subtask A** - Textual Content: tweets must be classified as reclaimed or not-reclaimed using the textual content of the tweet exclusively. Participants can decide to work on one or more of the following languages: Italian (A1), Spanish (A2), and English (A3).

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

[†]These authors contributed equally.

✉ sravisconti@gmail.com (S. Visconti); federica.manzi@unito.it (F. Manzi)

🌐 <https://github.com/Sara-Vis> (S. Visconti); <https://github.com/fede-m> (F. Manzi)

🆔 0009-0008-0618-5356 (F. Manzi)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

- **Subtask B** - Contextual Content: in addition to the textual content of the tweet, participants can also use the biography of the author of the tweet (when available) as extra contextual content to make the prediction. For this subtask, only tweets in Italian (B1) and Spanish (B2) were available.

Our team participated in both subtasks: for subtask A, we focused on Italian (A1) and English (A3), whereas for subtask B only on Italian (B1). For each subtask and language, we adopted different models and approaches.

After a brief description of related work in Section 2, we describe these different approaches in Section 3. Section 4 presents the results divided by subtask and language. Finally, Section 5 discusses results and presents an error analysis, while Section 7 concludes the paper.

2. Background

In this section, we provide background on both the theoretical and computational aspects of our work. The first part introduces the theoretical framework of reclamation as a linguistic phenomenon and reviews previous attempts to identify reclaimed language in NLP. The second part focuses on technical background, including the main techniques, models, and architectures used in the MultiPRIDE task.

2.1. Theoretical Background

Reclamation of slurs refers to the phenomenon in which members of a targeted, typically minority group intentionally use slurs that out-group members have historically employed to harm them, but for non-derogatory purposes. In such cases, the same word or utterance can serve as either an insult or a marker of solidarity, empowerment, or protest, depending on the specific context in which it is produced. A prominent example is the reclamation of the term "queer" by the LGBTQIA+ community at the end of the 20th century. As explained in detail in [9], activist groups adopted the slur "queer", which had acquired a strongly negative and derogatory connotation in the course of the century, and overturned its discriminatory power into an aggressive and powerful tool of protest and political identity.

The mechanisms underlying such reclaiming uses have been analyzed from different theoretical perspectives. Following the *echoic* perspective proposed in [10], in-group members of a community may echo expressions that are derogatory against their own group, in contexts that signal dissociation from their original offensive meaning, often producing an ironic effect. Furthermore, [11] discusses the practice of *reading*, common in drag queen communities, in which slurs are used against in-group members as a way of forming a "thick skin" and learning how to promptly reply when those same slurs are used with harmful intent by out-group members.

All these accounts highlight how context and pragmatic cues are crucial for determining whether a slur is used in an offensive or reclaiming way. This is particularly relevant in the context of content moderation on social media. For example, [6] evaluates the levels of toxicity attributed to Twitter posts from drag queens compared to those from white nationalist accounts using the Perspective API¹, a system developed by Jigsaw for automatic content moderation. Their results show that tweets from drag queens were assigned higher toxicity scores, due to the higher frequency of slurs like "f*g" and "b*tch", as well as neutral terms like "gay" or "lesbian", highlighting the inability of such systems to account for context and accurately distinguish reclaiming usages. Similar results were also reported in [12], where the authors compare different commercial content moderation systems and note how they all show the tendency to over-moderate marginalized groups, including the LGBTQIA+ community.

Although the task remains largely underexplored, recent NLP work has started to investigate reclaimed language and, more broadly, the distinction between abusive and non-abusive slur usage depending on context. A notable example in the Italian setting is the ReCLAIM project [7]. The authors leveraged the HODI dataset [13], which contains tweets annotated for hate speech, filtered the dataset for tweets including slurs commonly directed at the LGBTQIA+ community, and re-annotated them as

¹<https://perspectiveapi.com/>

reclaimed or not-reclaimed. The resulting dataset was then used to evaluate the zero-shot capabilities of an LLM to classify tweets as reclaimed or not. Furthermore, other relevant works on this topic are [14], [15], [16], and [17].

2.2. Technical Background

From a more technical perspective, the systems presented in the current work build upon several recent advances in transformer architectures and parameter-efficient fine-tuning. In this sub-section, we provide background on the key techniques and models we employ, including SetFit [1], DeBERTa [18], LoRA [4], and Mixture-of-Experts-based large language models such as the LLaMA 4 family.

Setfit (Sentence Transformer Fine Tuning) is a few-shot learning framework [19] that fine-tunes pre-trained sentence transformer models [20] to create a task-specific embedding space using contrastive learning. During training, triplets are formed, each consisting of a pair of training instances and a label indicating class similarity (1 if the examples belong to the same class, 0 otherwise). The model learns embeddings that bring pairs belonging to the same class closer while maximizing the distance between instances of different classes. This embedding space can then be used with a lightweight classifier, such as logistic regression, to perform classification. Being a framework for few-shot learning, the main advantage of using SetFit is that it is suitable for small or imbalanced datasets, as in our case, where the minority class contains relatively few examples.

The DeBERTa architecture, introduced in [18], builds on BERT and RoBERTa by introducing two new elements. The first is disentangled attention, which consists of representing tokens using two separate vectors: one encoding their content and the other their position. Attention weights among words are also disentangled: instead of having a single score per token-pair (encompassing both content and position information), the model computes three separate scores, namely content-to-content, content-to-position, and position-to-content. This enables the model to capture word order and relationships between tokens better. The second major novelty is an enhanced mask decoder, which improves the model’s ability to learn representations for masked tokens by leveraging positional information to make the prediction.

LoRA (Low-Rank Adaptation) is a parameter-efficient technique for fine-tuning models on downstream tasks by learning a small number of additional trainable parameters while keeping the original model weights frozen. Instead of updating the full transformer parameters, LoRA learns low-rank matrices that approximate the required weight updates and are added to the original weights. This allows for fine-tuning models at reduced computational costs, making it efficient, especially as the number of model parameters scales up.

Finally, the core idea of the Mixture of Experts (MoE) paradigm, introduced in [21], consists of decomposing a model into multiple specialized experts. For a given input, a routing mechanism selects the most relevant expert or group of experts, and only their corresponding parameters are activated to produce the final prediction. This allows for reducing computational and memory requirements and scaling to a large number of parameters while keeping inference and training costs manageable. We refer to [22] for a more in-depth overview.

More recently, MoE architectures have been successfully adopted in transformer-based LLMs, such as Switch Transformers [23], Mixtral [24], DeepSeek-MoE [25], and the LLaMA 4 family [5], which we use in our experiments. Specifically, LLaMA 4 Scout² has a total of 109B parameters distributed across 16 experts, which allows to activate only 17B parameters for each input. Together with LLaMA 4 Maverik³, which has 400B total parameters, 128 experts, and 17 B active parameters, LLaMA Scout was obtained by distilling from LLaMA 4 Behemoth, a much larger teacher model with approximately 2T parameters, 16 experts, and 288 B active parameters.

²<https://huggingface.co/meta-llama/Llama-4-Scout-17B-16E>

³<https://huggingface.co/meta-llama/Llama-4-Maverick-17B-128E>

3. Description of the System

In this section, we first describe the data and preprocessing steps and then the systems developed for the subtasks we participated in: subtask A in Italian (A1) and English (A3), and subtask B in Italian (B1). For each task and language, we adopted different architectures and modeling approaches, including an ensemble-based classifier and a two-step pipeline.

3.1. Data and Preprocessing

Table 1

Distribution of training data across subtasks, languages, and classes.

Subtasks	Language	Class 0	Class 1	Total
A, B	Italian	879	207	1086
A	English	938	88	1026

For the MultiPRIDE shared task, the organizers initially released a labeled training set for each language. The datasets for both subtasks were the same, except that the author’s biography could not be used for subtask A. Each instance consisted of a tweet, the author’s biography (available only for Italian in subtask B), the language, and the annotated binary label. The label indicated whether potentially offensive or neutral words and expressions targeting the LGBTQIA+ community in the tweet were used in a reclaiming (label 1) or non-reclaiming way (label 0).

It is crucial to stress that tweets belonging to label 0 included not only explicitly offensive tweets, but also instances that, even if not offensive or even positive, could not be classified as reclaimed.

Table 1 reports the distribution of each class per language. For both languages, the majority of instances belong to class 0, i.e., not-reclaimed, making both datasets imbalanced. This imbalance is even more pronounced in the English dataset, where class 1 accounts for only 8% of the dataset. To provide a more detailed overview of the linguistic properties of the data, Figure 1 shows the top 10 most frequent unigrams extracted from tweets in each class for English and Italian. The English dataset reveals a considerable lexical overlap between the two classes: the six most frequent unigrams, including the slur "f*ggot" and terms like "gay" and "queer", appear in both reclaimed and not-reclaimed tweets. This indicates that, in English, class distinction cannot rely on simple differences in the vocabulary. Conversely, Italian shows a much clearer separation of words across classes: class 1 is strongly associated with community-specific self-identifiers and markers of pride (e.g., "forci", "fr*cia", "l*lla", "pride"), whereas class 0 clusters around terms related to general discourse on transphobia and gender (e.g., "trans", "donne", "transfobia").

Furthermore, the log-odds ratios in Figure 2 highlight the most informative words for each class and confirm this trend. In Italian, the words most strongly correlated with each class are relevant to the reclamation topic, reflecting the same thematic difference (community self-identifications vs transphobia and gender) observed in unigrams. In contrast, English words for both classes are very generic and not relevant to the topic (e.g., "disease", "server", "everyone"). Based on these observations, we hypothesize that classifying tweets in Italian is likely to be more straightforward than in English, as clearer linguistic patterns can be identified.

Regarding data preprocessing, different strategies were adopted for each subtask and language. For Italian in subtask A, the data were kept unchanged. For English, we experimented with several approaches to try to mitigate the severe class imbalance. In particular, we attempted to undersample the majority class (class 0) or to upsample the minority class (class 1). Undersampling was applied by randomly selecting a subset of class 0 instances equal in size to the number of class 1 instances. We also experimented with larger subsets corresponding to 2:1 and 3:1 ratios of class 0 to class 1.

Upsampling was implemented using two strategies. The first consisted of simply duplicating each class 1 tweet in the training set. The second and more elaborate approach relied on data augmentation using the *llama-4-scout-17b-16e-instruct* model. For each tweet classified as reclaimed, we asked the LLM

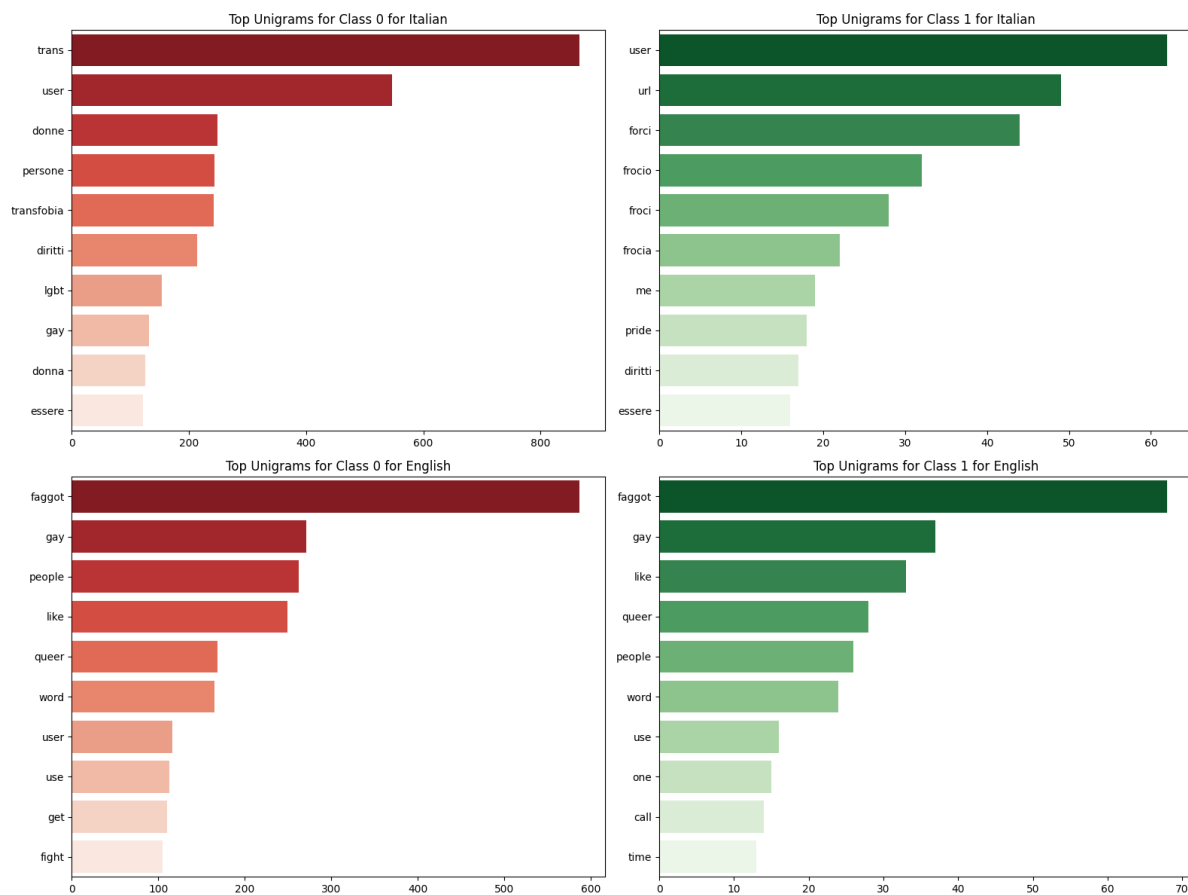


Figure 1: Comparison of the top 10 most frequent unigrams by class between Italian (top) and English (bottom). The red bars represent the not-reclaimed class (class 0), while the green bars represent the reclaimed class (class 1). Note the significant lexical overlap in English, where the top 6 unigrams are the same for both classes, compared to the distinct separation in Italian.

to generate two new paraphrased versions of the tweet. Since we did not want to give the model the responsibility to regenerate correct reclaimed versions (potentially changing offensive terms that could have changed the label of the tweet), we crafted a fixed list of words that had to stay the same in the paraphrased tweets. This ensured that the model only changed the non-crucial parts with no influence on the label of the tweet.

Finally, for subtask B, the author’s biography was concatenated to the tweet text. To distinguish the biography from the tweet, it was introduced with the string “BIO” at the beginning, allowing the model to incorporate additional contextual information when making predictions.

3.2. Models and Approaches

As discussed above, we used different modeling approaches depending on the language and subtask. In the following subsections, we describe each system architecture and training strategy in detail.

3.2.1. Subtask A1 - Textual Content in Italian

For subtask A1, we proposed an ensemble system (Figure 3) composed of three models employed as separate judges: a logistic regression classifier, a SetFit model [1], and an encoder-only transformer-based model (XML-RoBERTa [2]). Each model was trained separately before being combined into the ensemble.

The logistic regression model was trained on TF-IDF embeddings of the tweets. Each tweet was

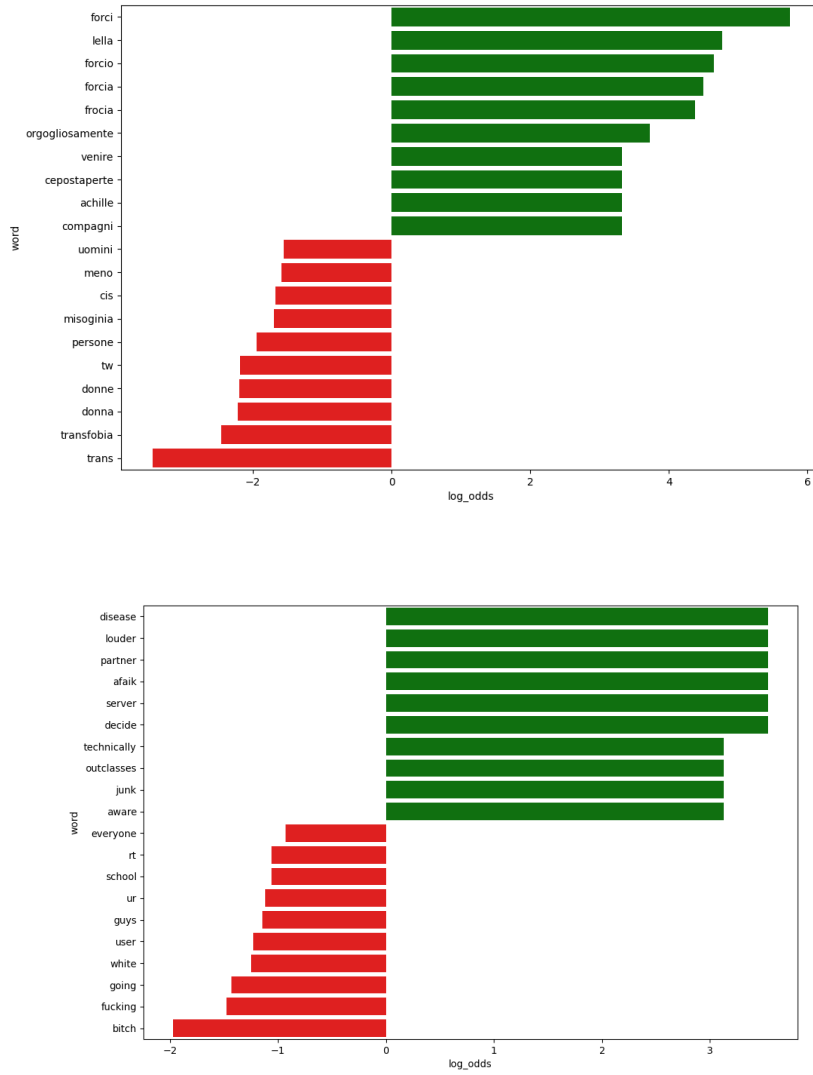


Figure 2: Comparison of the log-odds ratios of the top informative words for Italian (top) and English (bottom). Positive values (green bars) indicate words most strongly correlated to the reclaimed class (Class 1), while negative values (red bars) indicate words associated with the not-reclaimed class (Class 0). The Italian words show a strong semantic connection to the LGBTQIA+ community topic (e.g., *forcio*, *lella*), whereas the English words are very generic and do not have any semantic relevance to the topic.

encoded as a 6000-dimensional vector, corresponding to the 6000 most frequent unigrams and bigrams in the training set. Each vector entry represented the TF-IDF weight of the corresponding unigram or bigram in that tweet, reflecting its importance in the tweet relative to the entire training corpus.

For logistic regression, we used the *scikit-learn* implementation [26] and set the regularization coefficient C to 1, providing a moderate level of L2 regularization that balances underfitting and overfitting. To address the class imbalance, we set the *class weight* parameter to *balanced*, which automatically scales the weights inversely proportional to class frequencies, ensuring that the minority class has higher influence during training. The maximum number of iterations was increased to 1000 to ensure convergence, given the high dimensionality of the input vectors. The number of features and the n-gram range of TF-IDF, together with the regularization coefficient C were selected using grid search. All other logistic regression parameters were kept at their default values.

As mentioned in Section 2, the SetFit framework is composed of two steps: the first involving Sentence Transformers to embed the input text into a task-specific learned vector space, and the second relying

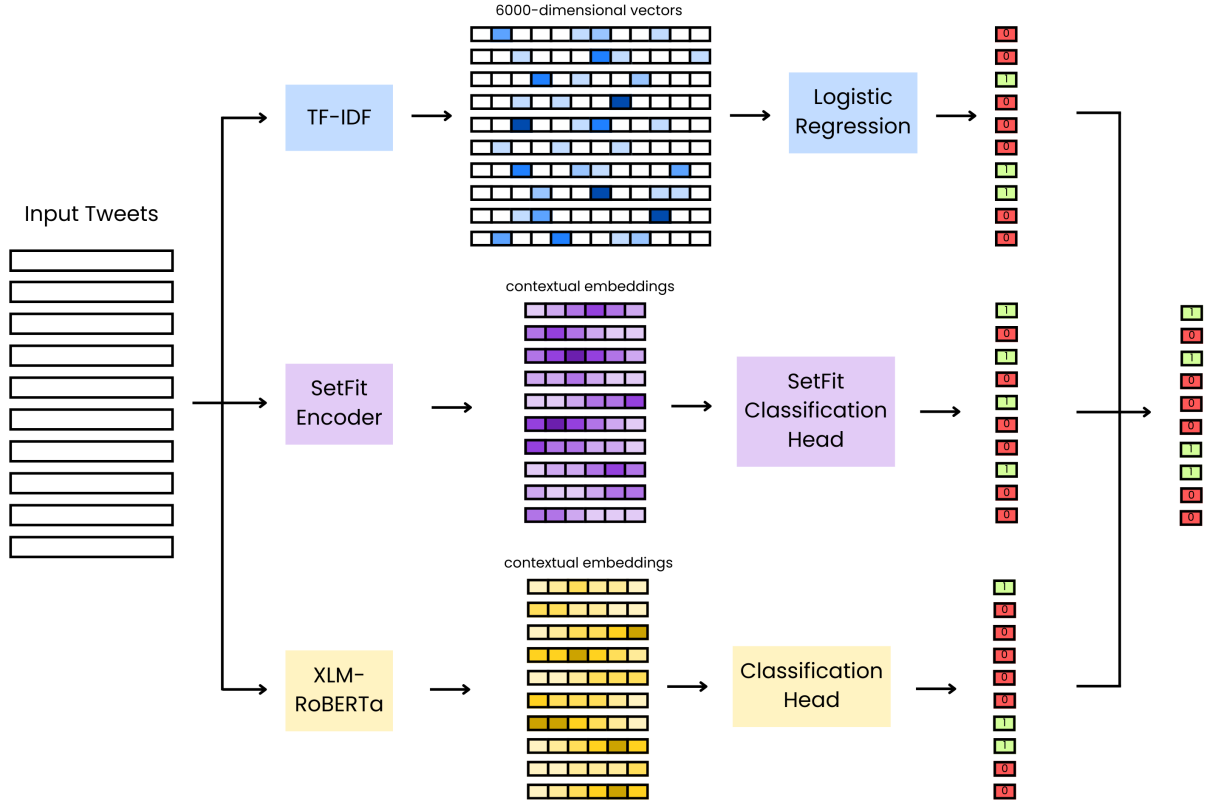


Figure 3: Ensemble system composed of three models employed as separate judges: logistic regression classifier, SetFit model, and encoder-only transformer-based model, proposed for subtask A1.

on a lightweight classification head to output the final prediction. In our experiments, embeddings were generated using *paraphrase-multilingual-MiniLM-L12-v2*⁴. Training was carried out for 10 epochs with a batch size of 8, setting the sampling strategy to undersampling to deal with class imbalance and reduce the number of instances of the majority class. As the classifier, the default logistic regression head was retained.

It is important to note that this approach differs from simply feeding a logistic regression classifier with sentence transformer embeddings, which we also experimented with, because SetFit focuses on learning task-specific embeddings rather than simple semantic representations.

For the third judge of the ensemble system, we fine-tuned XLM-RoBERTa [2], the multilingual version of the RoBERTa transformer model [27], on the full training set for 4 epochs and with a batch size of 8, with a learning rate of $1.03e-5$, and a weight decay of 0.06. We performed hyperparameter optimization using Optuna [28]. We used the default cross-entropy loss function and the AdamW optimizer. The maximum sequence length was set to 128 tokens, as tweets are generally short and do not require the full default length of 512 tokens.

At inference time, each tweet was first classified by the three models (logistic regression, Setfit, and XLM-RoBERTa) independently. The final prediction was then obtained via majority voting, by looking at the class predicted by at least two of the three models.

3.2.2. Subtask A3 - Textual Content in English

For Subtask A3 (English), the severe class imbalance in the English dataset posed a major challenge. As discussed in 3.1, there was no clear semantic and lexical distinction between words from different classes, making it difficult for models to learn meaningful patterns. To mitigate this effect, we tried

⁴<https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2>

different approaches for both data preprocessing and problem modeling. First, we undersampled the majority class (class 0) to contain only twice as many instances as class 1. The resulting dataset consisted of all 88 examples of class 1 and only 176 examples of class 0.

The final system we proposed is a two-step pipeline showed in Figure 4. In the first step, we used the DeBERTaV3 pre-trained transformer model [3], based on the DeBERTa architecture presented in 2, which we fine-tuned on the undersampled dataset using Low-Rank Adaptation (LoRA). This approach not only reduces computational and memory costs, but is also useful in few-shot scenarios or imbalanced datasets, such as the English subtask considered here. Training was carried out for 10 epochs with a batch size of 16, a learning rate of 1e-3, and a weight decay of 0.01. No class weights were needed due to the undersampling strategy.

For LoRA, we set the rank to 8: we deliberately chose a small value to limit how much model weights could change and reduce the risk of overfitting given the size of the dataset. This choice was motivated by early experiments, in which full fine-tuning of DeBERTa without LoRA led to overfitting. Moreover, the scaling factor alpha was set to 16 and dropout to 0.1 for regularization. At inference, the logits produced by DeBERTaV3 in the output layer are converted to probabilities using softmax. The confidence of the prediction is defined as the maximum probability across classes. We defined a threshold of 0.9, so that if the confidence falls below this threshold, the prediction is considered uncertain and the tweet undergoes the second step of the pipeline for further analysis.

The second step leveraged LLaMa 4 Scout, a powerful large language model based on a Mixture-of-Experts (MoE) architecture, to reclassify tweets for which the deBERTaV3 prediction was uncertain, i.e., whose confidence was falling below the 0.9 threshold. In the prompt fed to LLaMa 4, the task was framed as a three-class classification problem. Class 1 corresponded to reclaimed tweets, while the non-reclaimed class was subdivided into class 0, which included openly offensive tweets, and class 2, which covered neutral or even positive tweets towards the LGBTQIA+ community that are not inherently reclamatory.

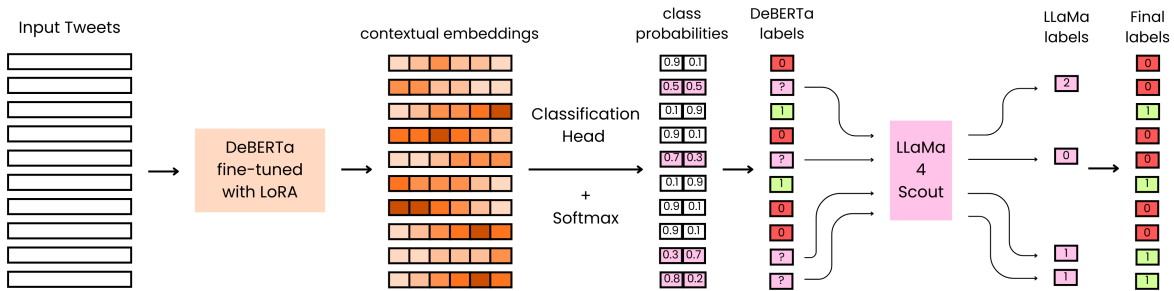


Figure 4: Two-step pipeline: pre-trained transformer model fine-tuned with LoRA, followed by an LLM to reclassify uncertain predictions, proposed for subtask A3.

The model received as input the original tweet together with the DeBERTaV3 prediction and was asked to judge whether the prediction of the smaller model was correct, and to provide an appropriate final label (0, 1, or 2). In a post-processing step, predictions assigned to class 2 were mapped to class 0 to match the binary label scheme of the task. The model was used in a zero-shot setting, meaning that no labeled tweets or in-context examples were provided in the prompt. Instead, the LLM was guided in its prediction by a set of rules, such as identifying first-person pronouns paired with slurs as markers of in-group membership and thus of reclaimed usage, as well as criteria to distinguish offensive from neutral content. To serve the LLM (specifically *llama-4-scout-17b-16e-instruct*), we used Groq⁵, a specialized hardware and software platform designed for low-latency inference. Additionally, we employed the Python *instructor* [29] library to enforce a structured JSON output, so that we did not have to deal with free text.

During the experimentation phase, we conducted several ablation studies for both steps of the

⁵<https://groq.com/>

pipeline. For step 1, we evaluated alternative classifiers, such as SetFit and logistic regression. Both approaches resulted in a recall stuck at 0.5, indicating that the models were predicting the majority class for all instances. Moreover, the output probabilities for both models were very similar to each other and therefore not very informative: logistic regression produced probabilities concentrated around 0.5, while SetFit around 1.0. As a result, it was not possible to establish a meaningful confidence threshold for further refinement with an LLM.

For step 2, we experimented with using LLaMa 4 Scout to directly predict all tweets, without the DeBERTa pre-filtering step, but this yielded worse results. Even with the pre-filtering step, we found that framing the task as an open classification task instead of as a correction of the output of DeBERTa also produced worse results. Other open-source LLMs were tested in Step 2, but LLaMa 4 Scout consistently outperformed alternatives, including LLaMa 4 Maverik and Qwen3 [30].

3.2.3. Subtask B1 - Contextual Content in Italian

For this subtask, we adopted the same ensemble architecture used for subtask A1, consisting of logistic regression, SetFit, and XLM-RoBERTa.

As described in Section 3.1, the only change concerned the input, where the tweet text was concatenated with the author’s biography. The rationale behind this choice was to allow the model to leverage potentially informative cues present in user bios, such as self-identifiers and markers of pride that may signal membership in the LGBTQIA+ community.

While this approach does not introduce any substantial subtask-specific architectural change, we believe it provides a simple and straightforward baseline for evaluating the contribution of the bio information without introducing additional complexity. More sophisticated strategies, such as explicitly predicting the author’s community membership and using this information to guide tweet classification, would require defining labels for LGBTQIA+ membership, which may be highly subjective and difficult to determine automatically. Although some bios contain explicit membership cues (e.g., pride flags) and tweets labeled as reclaimed are assumed to be authored by community members (following the assumption also used in [7]), explicitly modeling membership without relying on true labels could introduce additional bias and subjectivity. For this reason, we chose to rely solely on the provided labels and to supply the model with the biography as mere additional context to learn from.

4. Results

As mentioned earlier, our team participated in subtask A for Italian (A1) and English (A3), and subtask B for Italian only (B1). The test sets were released after the experimental phase without true labels. For each subtask, we applied our systems on the corresponding test data, collected the predicted labels and submitted them to the task organizers for evaluation.

Although up to two runs with different architectures and systems could be submitted for each subtask, we submitted a single run per task. Thus, only one set of results is available for each task.

The official ranking for the shared task was based on the macro F1 score. Accordingly, Table 2 reports macro-averaged precision, recall, and F1-score. Among the three subtasks, we can note that subtask A3 (English) exhibits the lowest performance across all metrics. This observation is consistent with the hypothesis we formulated during data analysis in Section 3.1, where we highlighted the lack of clear lexical and semantic patterns distinguishing the two classes in the English training set.

Furthermore, the results for subtask B1 are lower than those for A1, despite using the same system architecture. This suggests that simply including additional contextual information from the author’s biography, without adapting the underlying system accordingly, did not improve performance and may have even introduced noise.

Table 2
Results per subtasks

Subtask	Language	Precision (macro)	Recall (macro)	F1-Score (macro)
A1	Italian	0.9263	0.8620	0.8895
A3	English	0.6260	0.6412	0.6329
B1	Italian	0.9251	0.7664	0.8164

5. Discussion

The results highlight differences in the ability of systems to distinguish reclaiming from non-reclaiming usages across both languages and subtasks.

The linguistic properties of the data can largely explain the difference in performance between English and Italian in subtask A. The lexical analysis performed in Section 3.1 reveals that Italian exhibits strong correlations between specific vocabulary (e.g., community self-identifiers) and the reclamation label, making it easier to distinguish between classes by simply examining the linguistic features of the tweets. This made it possible to employ an ensemble system composed of smaller models such as Logistic Regression, SetFit, and XLM-RoBERTa for subtask A1.

In contrast, the English dataset exhibits high vocabulary overlap between classes, with the same frequent terms appearing in both classes. These different characteristics of the two languages help explain why classification in Italian (A1) achieved better performance than in English (A3), where distinguishing reclamation could not rely on explicit linguistic cues.

A further challenge was represented by class imbalance, which, although present in both datasets, was particularly severe in the English setting, where reclaiming tweets (class 1) made up only 8% of the entire dataset. Furthermore, unlike Italian, where this imbalance could be partially mitigated by a clearer semantic and lexical separation between classes, English could not rely on such linguistic distinctions. For several of the systems and models we experimented with, including SetFit and logistic regression, this resulted in recall values stuck at 0.5, with the models predicting the majority class for every instance. Moreover, the confidence scores associated to these predictions were tightly clustered around 1 for SetFit and 0.5 for logistic regression, making them unsuitable for confidence-based refinement strategies.

For these reasons, a more complex solution was required for the English subtask, motivating the adoption of the proposed two-step pipeline. Among the tested models, DeBERTaV3 fine-tuned with LoRA was the only configuration that achieved a recall above 0.5, indicating that it was able to learn patterns rather than just collapsing to predicting the majority class. Additionally, unlike simpler models, its confidence scores were sufficiently diverse to allow for the introduction of a threshold and the consequent second LLM-based refinement step.

In this second step, tweets for which DeBERTaV3’s prediction fell below the confidence threshold were re-evaluated using a more powerful model. Using an LLM allowed us to provide a more detailed task description in natural language and clear rules and patterns to guide the model’s decision process. Moreover, introducing a "neutral" class (class 2) helped the model distinguish actually reclaiming tweets from the ones whose content was neutral or even positive toward the LGBTQIA+ community but not inherently reclaiming. This allowed to reduce the number of false positives for class 1 in ambiguous cases. Interestingly, we also noticed that framing the task as a correction of the output of a smaller model yielded better results than directly asking the LLM to perform open classification of the tweets.

When comparing results across subtasks, we can observe a decrease in performance for subtask B1 compared to subtask A1. This suggests that simply concatenating additional contextual information about the author to the tweet text may introduce noise and worsen performance, rather than improve it, when such information is not explicitly modeled. This result highlights the need for a more structured approach to incorporate such information, ensuring that it contributes meaningfully to the prediction instead of introducing noise. However, our experiments also highlighted that determining whether an author belongs to the LGBTQIA+ community, and therefore whether they can use slurs and offensive

terms in a reclaiming way, is a highly non-trivial task, even for human annotators.

6. Error Analysis

The following error analysis highlights both some of the limitations of our approach and inherently ambiguous cases that are difficult to resolve automatically. For subtask A1 (Italian), the ensemble system sometimes struggled to distinguish between reclaiming and non-reclaiming uses of pejorative words such as "fr*cio", even when they co-occurred with first-person pronouns or markers of pride.

Tweet: "Che poi non è tanto per **noi fr*ci**, è che non so davvero come si possa esultare per aver negato tutele e diritti a delle persone disabili"

Translation: "It's not even that much about **us f*ggots**, I just really don't understand how you can cheer for denying protections and rights from disabled people"

Comment: Here, the slur in the tweet is clearly used in a reclaiming way, since the pronoun "us" identifies the author as belonging to the LGBTQIA+ community. However, our model did not label this tweet as reclaimed.

In contrast, there were instances where relying solely on these linguistic markers could be misleading. For example, the author of a tweet specifies that they belong to the LGBTQIA+ community, but they are offended by slur words such as "fr*cio" or "r*cchione". In this case, the non-reclaiming nature of the tweet depends on its content rather than the presence of in-group markers, since the author explicitly rejects their reclaiming use, making classification challenging.

Tweet: "@USER Io sono orgoglioso di essere gay. Non mi offendo se mi chiami gay. **Mi offendo** se mi chiami fr*cio, f*nocchio, r*cchione, ch*cca e chi più ne ha più ne metta"

Translation: "@USER I'm proud of being gay. I won't be offended if you call me gay. **I'll get offended** if you call me f*ggot, q*eer, s*ssy, and so on and so forth."

Comment: Here, there are markers of pride that help identify the author as part of the LGBTQIA+ community. However, the author says that they get offended if someone uses slurs towards them. Thus, the slurs they wrote in the tweet were not written to reclaim them, making the tweet not-reclaimed. Our system labeled this instance as reclaimed.

This shows that, despite generally achieving better performance on Italian than English, simpler models alone may not suffice for more complex cases, where more reasoning capabilities are needed to interpret the context correctly. However, some ambiguous cases like the one in the example remain challenging, possibly even for humans.

For subtask A3 (English), errors reveal additional challenges. The annotation appears less consistent than in Italian, and it is often unclear why a particular tweet was labeled as reclaimed (class 1) rather than not-reclaimed (class 0). Some annotation rules, such as requiring the author to belong to the community for reclaiming usage, seemed less consistently applied, especially since the English dataset did not include author biographies. This increases subjectivity and makes it harder for models to learn clear patterns to distinguish between classes.

Tweet 1: "F*ck Louis CK and any straight d*uchebag who feels legitimate calling someone a "f*ggot" because of his bullshit stand-up. Comedians don't get to decide what is acceptable and what is not."

Tweet 2: "Sometimes I think that being gay is completely accepted in our society and then I'm quickly reminded by someone at another table that I'm a "f*ggot""

Comment: Here, we cannot be certain whether the tweets’ authors belong to the LGBTQIA+ community. However, they are both reporting cases in which the same slur was used in a non-reclaiming way by someone who is not part of the community. Thus, one might want to classify these tweets as not-reclaimed (label 1). Nevertheless, one might assume the authors are indeed part of the community, since they seem knowledgeable enough on the topic to avoid using the slurs when not belonging to the community. In this scenario, they might have decided to use the exact slur that was used against them to reclaim it. Therefore, the reclaimed class (label 0) would also be appropriate. In the dataset, the first tweet had label 1 while the second had label 0.

Furthermore, the limited performance observed even when using more powerful LLMs suggests that simply increasing model capacity may not necessarily lead to better results. Even with prompting, the rules provided to distinguish between classes need to be aligned with the annotator’s perceptions and annotation guidelines for them to be effective. This emphasizes the inherent difficulty of the task and its leaning towards subjectivity and indicates that providing clearer annotation guidelines could benefit automatic classification approaches.

Overall, this analysis highlights that the task is challenging even for humans, particularly when linguistic cues are subtle or the annotation schema is ambiguous, limiting the effectiveness of automatic systems, even for powerful LLMs. For LLMs to work effectively, clear and detailed instructions are essential. However, providing such specific instructions is challenging in a task that is inherently subjective and open to interpretation, as it requires an in-depth description of how to handle diverse and nuanced cases.

7. Conclusion and Future Work

In this work, we described our participation in the EVALITA 2026 MultiPRIDE task. This task focused on classifying Italian and English tweets containing words and slurs potentially offensive to the LGBTQIA+ community, distinguishing between reclaiming and non-reclaiming uses. The task proved difficult due to a series of challenges. On a technical level, the high class imbalance was problematic, especially for the English dataset. On a theoretical level, the task is inherently difficult, as determining what counts as reclaiming is subjective and not always consistent.

Our experiments showed that Italian tweets (subtask A1) were easier to classify due to clearer lexical and semantic distinctions between classes, allowing simpler models to perform well, without the need for powerful reasoning models. In this case, using an ensemble model composed of logistic regression, SetFit, and XLM-RoBERTa with majority vote was sufficient to achieve decent performance. In contrast, English tweets (subtask A3) presented further challenges, due to the lack of clear linguistic clues to distinguish the reclaimed from the not-reclaimed class. This required the usage of a more complex two-step pipeline, which leverages DeBERTaV3 with LoRA fine-tuning as a first pre-filtering step, and a more powerful model, LLaMa 4 Scout, in the second step to refine low-confidence predictions. However, despite using powerful LLMs, classification remained difficult due to annotation subjectivity, ambiguous tweets, and the limited availability of contextual information in English.

Further work could explore more structured methods for incorporating author context (subtask B1), a more balanced dataset, especially for English, and clearer annotation guidelines that can be used to structure prompts and explain the task to LLMs more effectively. Overall, our study highlights the potential and limitations of using both traditional machine learning models and more powerful LLMs for nuanced and context-dependent detection of hate speech and offensive language in social media.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4o, Gemini 3 Pro, and Grammarly to correct grammar, spelling and improve the readability of the paper through sentence polishing and rephrasing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] L. Tunstall, N. Reimers, U. E. S. Jo, L. Bates, D. Korat, M. Wasserblat, O. Pereg, Efficient Few-Shot Learning Without Prompts, *arXiv* (2022). doi:10.48550/arXiv.2209.11055.
- [2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020*, pp. 8440–8451. doi:10.18653/v1/2020.acl-main.747.
- [3] P. He, J. Gao, W. Chen, DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing, 2021. *arXiv:2111.09543*.
- [4] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-Rank Adaptation of Large Language Models, in: *International Conference on Learning Representations, 2022*. URL: <https://openreview.net/forum?id=nZeVKeeFYf9>.
- [5] Meta AI, The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, 2025. URL: <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>.
- [6] T. Oliva, Fighting Hate Speech, Silencing Drag Queens? Artificial Intelligence in Content Moderation and Risks to LGBTQ Voices Online, *Sexuality & Culture* (2021). doi:10.1007/s12119-020-09790-w.
- [7] L. Draetta, C. Ferrando, M. Cuccarini, L. James, V. Patti, ReCLAIM Project: Exploring Italian Slurs Reappropriation with Large Language Models, in: *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024*, pp. 335–342.
- [8] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, MultiPRIDE at EVALITA 2026: Overview of the Multilingual Automatic Detection of Slur Reclamation in the LGBTQ+ Context Task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026*.
- [9] E. Nossem, Queer, Frocia, Femminiellə, Ricchione et al. – Localizing ‘Queer’ in the Italian Context., *Gender/Sexuality/Italy* 6 (2019) 1–27. doi:10.15781/31yc-ys20.
- [10] C. Bianchi, Slurs and Appropriation: An Echoic Account, *Journal of Pragmatics* 66 (2014) 35–44. doi:10.1016/j.pragma.2014.02.009.
- [11] S. McKinnon, “Building a thick skin for each other”: The use of ‘reading’ as an interactional practice of mock impoliteness in drag queen backstage talk, *Journal of Language and Sexuality* 6 (2017) 90–127. doi:10.1075/jls.6.1.04mck.
- [12] D. Hartmann, A. Oueslati, D. Staufer, L. Pohlmann, S. Munzert, H. Heuer, Lost in Moderation: How Commercial Content Moderation APIs Over- and Under-Moderate Group-Targeted Hate Speech and Linguistic Variations, in: *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, CHI ’25, ACM, 2025*, p. 1–26. doi:10.1145/3706598.3713998.
- [13] D. Nozza, A. T. Cignarella, G. Damo, T. Caselli, V. Patti, HODI at EVALITA 2023: Overview of the first Shared Task on Homotransphobia Detection in Italian, in: *EVALITA, volume 3473 of CEUR Workshop Proceedings, CEUR-WS.org, 2023*.
- [14] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate Speech Detection and Reclaimed Language: Mitigating False Positives and Compounded Discrimination, *Proceedings of the 16th ACM Web Science Conference (2024)*. URL: <https://api.semanticscholar.org/CorpusID:269228809>.
- [15] E. W. Pamungkas, V. Basile, V. Patti, Do You Really Want to Hurt Me? Predicting Abusive Swearing in Social Media, in: *International Conference on Language Resources and Evaluation, 2020*. URL: <https://api.semanticscholar.org/CorpusID:218974299>.
- [16] A. T. Cignarella, M. Sanguinetti, S. Frenda, A. Marra, C. Bosco, V. Basile, QUEEREOTYPES: A Multi-Source Italian Corpus of Stereotypes towards LGBTQIA+ Community Members, in: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024*, pp. 13429–13441. URL:

<https://aclanthology.org/2024.lrec-main.1176/>.

- [17] L. Veloso, L. Hirlimann, P. Wicke, H. Schütze, SLAYiNG: Towards Queer Language Processing (2025). doi:10.48550/arXiv.2509.17449.
- [18] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with Disentangled Attention, in: International Conference on Learning Representations, 2021. URL: <https://openreview.net/forum?id=XPZlaotutsD>.
- [19] H. Liu, D. Tam, M. Muqeeth, J. Mohta, T. Huang, M. Bansal, C. Raffel, Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning, arXiv (2022). doi:10.48550/arXiv.2205.05638.
- [20] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.
- [21] J. Robert, J. Michael, N. Steven, H. Geoffrey, Adaptive Mixtures of Local Experts, Neural Computation 3 (1991) 79–87. doi:10.1162/neco.1991.3.1.79.
- [22] S. Mu, S. Lin, A Comprehensive Survey of Mixture-of-Experts: Algorithms, Theory, and Applications, 2025. URL: <https://arxiv.org/abs/2503.07137>. arXiv:2503.07137.
- [23] W. Fedus, B. Zoph, N. Shazeer, Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, Journal of Machine Learning Research 23 (2022) 1–39. URL: <http://jmlr.org/papers/v23/21-0998.html>.
- [24] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, T. Le Scao, T. Gervet, T. Lavril, T. Wang, T. Lacroix, W. El Sayed, Mixtral of Experts, 2024. doi:10.48550/arXiv.2401.04088. arXiv:2401.04088.
- [25] D. Dai, C. Deng, C. Zhao, R. X. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, Z. Xie, Y. K. Li, P. Huang, F. Luo, C. Ruan, Z. Sui, W. Liang, DeepSeekMoE: Towards Ultimate Expert Specialization in Mixture-of-Experts Language Models, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 1280–1297. doi:10.18653/v1/2024.acl-long.70.
- [26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine Learning in Python, 2018. URL: <https://arxiv.org/abs/1201.0490>. arXiv:1201.0490.
- [27] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, Huhhot, China, 2021, pp. 1218–1227. URL: <https://aclanthology.org/2021.ccl-1.108/>.
- [28] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A Next-generation Hyperparameter Optimization Framework, 2019. doi:10.48550/arXiv.1907.10902. arXiv:1907.10902.
- [29] J. Liu, Contributors, Instructor: A library for structured outputs from large language models, 2024. URL: <https://github.com/instructor-ai/instructor>.
- [30] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 Technical Report, 2025. doi:10.48550/arXiv.2505.09388. arXiv:2505.09388.

The Hate Busters at MultiPRIDE: Automatic Identification of Reappropriated Slurs in Multilingual LGBTQ+ Discourse

Aurora Ciminelli[†], Giulia Corvino[†], Camilla Gentili[†] and Marco Viviani^{*}

Università degli Studi di Milano-Bicocca, Dipartimento di Informatica, Sistemistica e Comunicazione (DISCo) – Edificio U14 (ABACUS), Viale Sarca, 336 – 20126 Milan, Italy

Abstract

This paper presents the methodology developed for the MultiPRIDE shared task by the Hate Busters team, which focuses on automatically identifying reappropriative intent in multilingual social media messages containing LGBTQ+ related slurs. Reappropriation is a nuanced sociolinguistic phenomenon in which historically derogatory terms are deliberately reclaimed with positive or community-affirming meanings. To model this behavior, we work with the multilingual dataset provided for the MultiPRIDE task, including texts in Italian, Spanish, and English. Our approach integrates a comprehensive methodological pipeline, encompassing preprocessing of noisy social media text, the construction of traditional sparse lexical representations such as Bag-of-Words and TF-IDF, and contextual Transformer-based embeddings from XLM-RoBERTa, followed by training linear classifiers and fine-tuned Transformer models. The goal is to systematically compare how different modelling paradigms capture the subtle and context-dependent characteristics of reappropriative language.

Warning: This paper contains examples of explicitly offensive content.

Keywords

Reappropriative intent, Semantic reclamation, LGBTQ+, Natural Language Processing, Text Classification

1. Introduction

Semantic reclamation is a linguistic phenomenon in which terms that have historically carried derogatory or offensive meanings are reappropriated by the communities they originally targeted, acquiring new, neutral, or empowering connotations [1, 2, 3]. While reclaimed terms may retain their offensive potential in certain contexts, their meaning can shift significantly depending on the speaker, audience, and communicative intent. This phenomenon is particularly evident in the use of slurs within the LGBTQ+ community, where words traditionally employed as insults are often used with a reclaiming intent to express identity, solidarity, or empowerment [4]. In online environments, especially on social media platforms, distinguishing between derogatory and reclaiming uses of such language becomes a critical challenge, as surface-level lexical cues are often insufficient to capture the underlying intent. It is within this context that MultiPRIDE at EVALITA 2026 [5] is situated. The MultiPRIDE task [6] focuses on identifying reclaimed slurs in Italian, Spanish, and English within content related to the LGBTQ+ community.

The approach proposed by the Hate Busters team within MultiPRIDE integrates a comprehensive methodological pipeline designed to tackle the challenges posed by the identification of reclaimed slurs in social media text. The pipeline begins with preprocessing techniques specifically tailored to handle the noisy and informal nature of user-generated content, including normalization, tokenization, and cleaning of textual data. Following preprocessing, we construct both traditional sparse lexical representations, such as Bag-of-Words and TF-IDF, and dense contextual embeddings derived from Transformer-based models, in particular XLM-RoBERTa. These representations are then employed

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

^{*}Corresponding author.

[†]These authors contributed equally.

✉ a.ciminelli1@campus.unimib.it (A. Ciminelli); g.corvino2@campus.unimib.it (G. Corvino); c.gentili3@campus.unimib.it (C. Gentili); marco.viviani@unimib.it (M. Viviani)

🌐 <https://ikr3.disco.unimib.it/people/marco-viviani/> (M. Viviani)

🆔 0000-0002-2274-9050 (M. Viviani)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

to train linear classifiers as well as fine-tuned Transformer models, allowing us to investigate the relative effectiveness of different modelling paradigms in capturing the subtle and context-dependent characteristics of reclamatory language. Our main contributions are the integration of a robust and flexible pipeline capable of handling multilingual social media data, the systematic comparison of sparse and dense text representations for the detection of reclaimed slurs, and the evaluation of both classical and state-of-the-art Transformer-based models. This approach highlights how different techniques can capture the nuanced meanings that arise in semantic reclamation and provides a structured methodology for future work on intent-sensitive classification tasks in the context of LGBTQ+ language use.

2. The MultiPRIDE Dataset and Tasks

This section provides information on the dataset released by MultiPRIDE, an analysis of these data, and an overview of the tasks proposed.

2.1. Description of the Dataset

The dataset provided by MultiPRIDE is a multilingual corpus composed of textual social media content in Italian, Spanish, and English, aggregated from heterogeneous data sources. As specified in the official MultiPRIDE guidelines [5], the data originate from:

- Italian: the *TWITA* collection, a large-scale corpus of Italian tweets [7];
- Spanish: the *LGBTQI+ Dataset 2020–2022*, consisting of tweets related to LGBTQ+ topics [8];
- English: a composite dataset drawn from Twitter, Reddit, and TV series transcripts focusing on reclaimed and potentially derogatory terminology [9].

All texts were collected from publicly available sources between 2020 and 2022.

2.2. Preliminary Analysis of the Data

The initial datasets, provided by the organizers of the MultiPRIDE challenge, were harmonized through a uniform filtering pipeline applied across the three languages. A first selection step was conducted via keyword-based filtering, using terms related to homosexuality and derogatory expressions derived from the *Hurtlex* lexicon (e.g., fag, gay, bitch, etc.). To further refine the corpus and increase the likelihood of retrieving instances of reclaimed language, a second filtering step targeted positively connoted and community-oriented terms, such as pride, queer, LGBT, and rainbow.

After the automated filtering phase, all candidate texts were manually annotated with a binary label indicating whether an LGBTQ+-related term was used with **reappropriative intent** (*yes*) or not (*no*).

The initial set of data includes 1,811 observations for Italian, 1,711 for English, and 1,461 for Spanish. An initial exploratory analysis of the data showed that, in all languages, the distribution of labels is not balanced, with non-reclaimed instances representing the majority class. As illustrated in Figure 1, which reports the label distribution computed by merging the three languages’ training sets, the dataset is strongly skewed toward the non-reclaimed category. This imbalance motivates the adoption of evaluation metrics that are robust to skewed label distributions.

We also examined the distribution of message lengths, measured in both tokens and characters. As expected for social media content, the majority of texts are relatively short; however, the presence of a long tail of longer messages was particularly informative for subsequent modelling choices. As shown in Figure 2, which reports the token-length distribution computed over the merged multilingual dataset, most texts fall between 20 and 60 tokens, with a clear concentration around the average length of 35.6 tokens. The maximum observed length is 148 tokens, resulting in a heavily right-skewed distribution. Since the choice of the maximum sequence length hyperparameter directly affects memory usage and the risk of losing information through truncation, this analysis was essential for setting an appropriate threshold. Therefore, the analysis of these properties supported different preprocessing decisions described later.

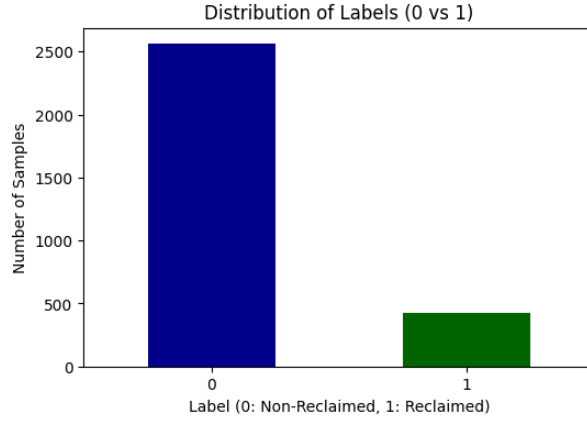


Figure 1: Distribution of labels in the merged multilingual training set (Italian, English, Spanish).

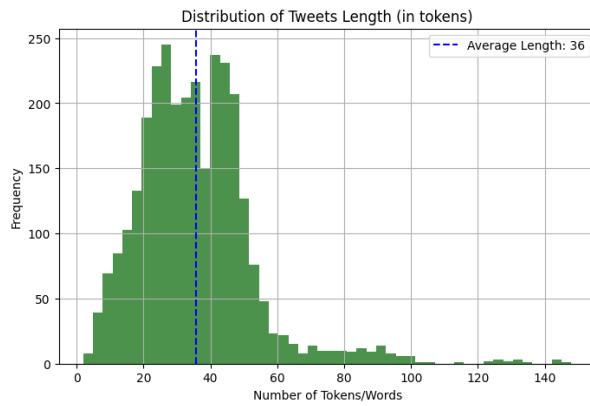


Figure 2: Distribution of tweet lengths (in tokens). The dashed line marks the mean length.

The training data are distributed in three distinct CSV files, one for each language. All files share a common structure defined by the following fields:

- **id:** Unique identifier of the message;
- **text:** Textual content of the message;
- **label:** Binary label (*yes* = reclaimed, *no* = not reclaimed);
- **bio:** Author biography, when available;
- **lang:** Language of the message.

In addition, contextual information about the author’s profile is available only for the Italian and Spanish datasets, where the *bio* field contains short biographical descriptions extracted from the user’s public profile. An inspection of missing values confirms this distribution: the *bio* field is absent for 100% of English instances, while missing rates are comparatively low for Italian ($\approx 8\%$) and Spanish ($\approx 6\%$) entries in the training data, resulting in an overall missing proportion of $\approx 39\%$ across languages. Similar patterns are observed in the test set. This additional information is specifically relevant to Task B, which permits the use of contextual metadata alongside the textual content in order to improve the classification of reappropriative intent.

A preview of the dataset structure is reported in Figure 3, which displays a subset of Italian instances and illustrates the format and content of the fields.

Finally, we also examined the lexical distribution of the corpus through word-cloud visualizations, distinguishing between the two labels (see Figure 4 and Figure 5). The word cloud associated with Label 0 highlights the prevalence of denigratory terms as well as high-frequency but non-informative tokens (e.g., *user*, *url*). On the other hand, the word cloud for Label 1 reflects the linguistic complexity

- **Task B** extends the task by allowing the use of contextual metadata about the author, specifically the biographical information provided in the *bio* field. This additional context can support the disambiguation between derogatory and reappropriative uses of language.

In this paper, we consider both tasks, and for each task, we provide the relevant details on our approach in the following Methodology section.

3. Methodology

This section illustrates the proposed methodology for addressing the tasks introduced by MultiPRIDE. In particular, it describes the data preprocessing phase, the textual representation techniques applied to the data, and the models selected for classification.

3.1. Text Preprocessing

We applied an identical initial preprocessing pipeline to the English, Italian, and Spanish datasets. This step aimed at normalizing the raw text and ensuring that each message was represented as a single, well-formed string before any task-specific processing.

First, the training data released by the challenge organizers were loaded from CSV files into pandas DataFrames using UTF-8 encoding. This choice guarantees a consistent treatment of multilingual content and avoids issues with non-ASCII characters, such as accented letters and symbols frequently present in Italian and Spanish tweets.

We then standardized the textual field corresponding to the message content. Since the original files contained heterogeneous line-break encodings (arising, for example, from different operating systems or from escaped sequences within the exported text), we explicitly normalized all newline patterns. By collapsing all these variants into a single space, we ensure that each message is stored as one continuous line of text, while preserving word boundaries. This is particularly important in our setting, where line breaks may appear both because of platform-dependent formatting and because of how the textual content was exported (e.g., embedded escape sequences in longer strings). Normalizing these artifacts reduces spurious token boundaries and facilitates consistent tokenization across languages.

In addition, we defined a regular expression pattern to identify emoji characters in the text. We used this pattern as a building block for subsequent stages of preprocessing and analysis, where emoji may be either removed, normalized, or explicitly modeled as features depending on the preprocessing strategy. Although language-independent, its early definition allows us to handle emoji in a uniform way in all three corpora. In our experiments, emojis were removed under the intensive preprocessing pipeline adopted for sparse lexical models (BoW/TF-IDF), while they were preserved and normalized under the mild preprocessing used for XLM-RoBERTa.

Overall, this shared preprocessing step produced a clean, line-break-normalized version of the text field for English, Italian, and Spanish, providing a homogeneous basis for the language-specific and task-specific processing described in the following sections.

3.1.1. Task A

For Task A, which relies solely on the textual content of the message, we first constructed a unified multilingual dataset by integrating the three monolingual corpora (i.e., Italian, Spanish, and English). The workflow of Task A is illustrated in Figure 6 and detailed below.

To obtain the integrated dataset, it was necessary to perform preliminary processing steps on the individual corpora in order to ensure their correct merging. For example, the Italian and Spanish datasets already include a *bio* field describing the author’s profile, whereas the English dataset does not. To obtain a coherent structure, we explicitly added an empty *bio* column to the English data and aligned the column order across all corpora. The three datasets were then concatenated into a single DataFrame.

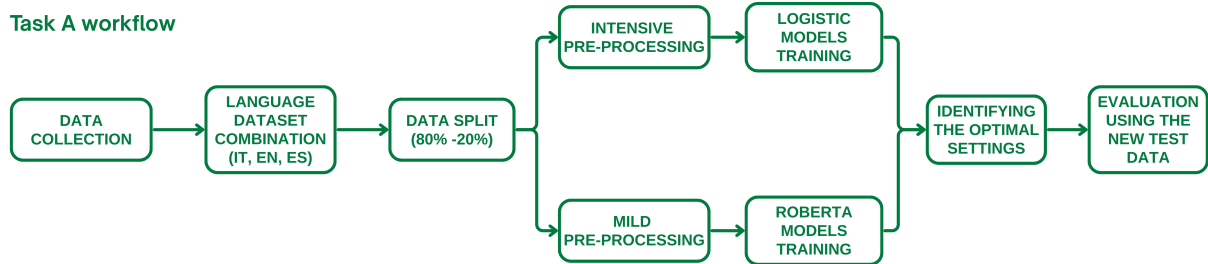


Figure 6: *The workflow for Task A.*

After concatenation, we performed basic sanity checks on the resulting dataset (e.g., verification of the overall shape and column names, and analysis of the label distribution) to ensure that instances from all three languages are correctly merged. This integration step allows us to train a single model over a multilingual corpus.

Starting from our integrated dataset, we applied a unified text preprocessing pipeline to all instances, parametrized by the language of the message. The goal of this pipeline is to normalize surface variation, remove noise typical of social media text, and obtain a lemmatized representation suitable for subsequent modeling.

We defined a set of language-independent cleaning functions, applied in a fixed order to each message. For the intensive preprocessing adopted for sparse lexical models (Bag-of-Words and TF-IDF), we started by removing all emoji characters using a regular expression covering the corresponding Unicode ranges. In contrast, under the mild preprocessing strategy used for XLM-RoBERTa, emoji characters were preserved and later converted into their textual descriptions. Next, hyperlinks, both in `http(s)://` and `www.` format, were removed. We then also removed Twitter-style mentions (strings starting with `@` followed by alphanumeric characters and hyphens), the conventional retweet marker `RT` at the beginning of a message (when present), all numeric sequences (in this task, numbers rarely contribute to detecting reclamatory usage and may interfere with tokenization and lemmatization), and finally all punctuation characters were replaced with whitespace to standardize token boundaries and reduce noise due to idiosyncratic punctuation.

The final step of the preprocessing pipeline consists of language-aware lemmatization combined with stopword removal. For each message, we rely on the `lang` field to select the appropriate language-specific processing rules (English, Italian, or Spanish). Starting from the cleaned text, the message is first segmented into tokens using simple whitespace and punctuation-based heuristics. Each token is then mapped to its corresponding lemma through language-specific morphological rules and lexicons. This procedure reduces inflected forms to a common base form (e.g., `reclaimed`, `reclaiming` → `reclaim`), thereby decreasing sparsity and allowing the model to capture lexical patterns that are invariant to tense, number, or gender.

In parallel, we define a multilingual stopword list by combining three sets of high-frequency function words, one for each language. During lemmatization, any lemma that appears in this multilingual stopword list is discarded. This filtering step removes items such as determiners, conjunctions, and auxiliary verbs, which carry limited semantic content for the reclamation detection task and would otherwise dominate the feature space.

The remaining lemmas are converted to lowercase and concatenated back into a single string, which is stored in a new column of the unified dataset. This lemmatized, stopword-filtered representation constitutes the final input used by our models in Task A and provides a compact, language-consistent basis for learning cross-lingual patterns associated with reclamatory and non-reclamatory uses.

3.1.2. Task B

While Task A relies exclusively on the textual content of the message, Task B additionally exploits contextual metadata about the author, specifically the biographical information provided in the Italian

and Spanish datasets. Because this field is heterogeneous, often noisy, and inconsistently encoded, an additional preprocessing stage was required to clean, normalize, and integrate the biography (bio) field with the corresponding message (text). The goal was to construct a reliable combined representation that could be used directly as model input. The workflow of Task B is illustrated in Figure 7 and detailed in the following.

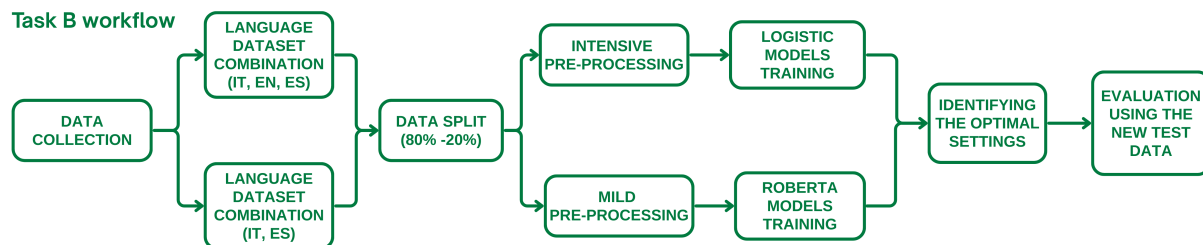


Figure 7: *The workflow for Task B.*

The preprocessing of Task B operates on both the message content and the author biography. Due to inconsistencies in the raw data, we first normalized line breaks and removed problematic encodings that hindered downstream processing. For both the text and bio columns, we applied a uniform sequence of cleaning steps. Since the datasets contain multiple encodings of line breaks, including literal escape sequences, Windows-style line breaks, and true newline characters. All these forms were systematically replaced with a single space. This ensures that both text and bio fields are represented as uninterrupted one-line strings, similar to the normalization applied in the initial preprocessing. Then, under the intensive preprocessing configuration adopted for sparse lexical models, a broad Unicode pattern was defined to filter out emoji and pictographic symbols from the textual content. Although emojis may signal sentiment or stance, in this setting, their presence was deemed potentially noisy, particularly in the biography field. In contrast, when fine-tuning XLM-RoBERTa, emojis were preserved and converted into their textual descriptions in order to retain potential pragmatic and contextual cues. Finally, both fields were explicitly converted to string type and stripped of superfluous whitespace, reducing variability and preventing alignment issues in concatenation.

The biography field required careful treatment due to the presence of multiple textual placeholders denoting missing content. Therefore, we converted known missing-like patterns into actual NaN values, and replaced all resulting NaN entries with empty strings. This choice prevents models from mistakenly interpreting placeholder text as meaningful linguistic content once concatenation is performed.

After these operations, the dataset contained no missing values in either field, ensuring that every instance yields a clean concatenated representation.

In order to enforce minimal surface-level uniformity and avoid spurious token boundaries after concatenation, a final cleaning function was applied to both fields, which collapsed multiple whitespace characters into a single space and removed leading and trailing whitespace.

For Task B, the core idea is to provide the model with augmented input that includes both the message content and the author’s contextual information. We therefore concatenated the cleaned biography to the cleaned message. Empty biographies simply contribute no additional content, while non-empty biographies enrich the textual signal with relevant contextual cues (such as explicit LGBTQ+ self-identification or references to activist roles). The resulting additional field constitutes the final input representation for Task B.

3.1.3. Validation Set

The validation set provided by the challenge organizers was subjected to the same preprocessing pipeline applied to the training data.

To recap, as for the training portion, all messages were normalized by removing URLs, user mentions, numbers, retweet markers, and punctuation, followed by lowercasing and whitespace reduction. Emoji characters were removed under the intensive preprocessing configuration adopted for sparse lexical

models, whereas they were preserved and normalized under the mild preprocessing used for XLM-RoBERTa. For Task A, each instance was then processed through the same multilingual lemmatization and stopword filtering procedure based on the language specified in the dataset. For Task B, the validation biographies underwent the same cleaning and harmonization steps as the training biographies, including newline normalization, removal of placeholder strings, and conversion of missing values into empty strings before concatenation with the message text.

By mirroring the full preprocessing workflow on the validation set, we ensure that the models are evaluated on data that reflect the exact same transformations applied during training, guaranteeing full methodological consistency.

3.2. Text Representation Models

In order to investigate the most effective textual representations for the tasks under consideration, we explored both sparse and dense representations, which are described below.

3.2.1. Sparse Models

To convert raw multilingual social media text into a format suitable for machine-learning algorithms, we employ embedding representations that capture both semantic and syntactic properties of language. Given the multilingual and socially varied nature of the dataset our embedding strategy prioritizes **cross-lingual comparability**, **sensitivity to lexical variation**, and **robustness to informal language** typical of social media dialogue.

Within this framework, we adopt **sparse lexical representations**, specifically Bag-of-Words (BoW) and Term Frequency–Inverse Document Frequency (TF–IDF).

Bag-of-Words provides a straightforward count-based encoding of the textual content, enabling models to capture the distribution of key terms associated with reappropriation without imposing assumptions on word order or syntactic structure. This representation is particularly suitable for social media text, where meaning is often conveyed through distinctive lexical choices, hashtags, or targeted slurs.

Building on this, **TF–IDF** further normalizes word frequencies by penalising terms that are overly common in the corpus and up-weighting more informative ones. This helps distinguish contextually important vocabulary, which is also especially relevant for detecting linguistic reclamation, where the interpretive weight of specific keywords varies across communities and speakers.

Finally, the combination of the BoW and TF-IDF models enables the preservation of a high-resolution view of the lexical surface while simultaneously mitigating noise from tokens that are either extremely frequent or of minimal informative value.

3.2.2. XLM-RoBERTa

In addition to the classical linear models built on Bag-of-Words and TF-IDF, we experimented with a more sophisticated, deep learning approach based on a multilingual Transformer encoder. XLM-RoBERTa (XLM-R) is a state-of-the-art multilingual Transformer trained on 100+ languages using masked language modeling [10]. It provides rich contextual embeddings, allowing the model to capture: cross-lingual semantic patterns, subtle lexical nuances, contextual reappropriation elements (that are difficult for BoW/TF-IDF to encode). The model has been fine-tuned to ensure optimal performance on the binary Reappropriation classification task.

Unlike the sparse lexical representations used in our baseline models (Bag-of-Words and TF-IDF), XLM-RoBERTa requires a different preprocessing strategy. Consequently, we applied a dedicated preprocessing pipeline designed to preserve as much of the original data as possible.

The preprocessing function defined for the XLM-R model performs a minimal set of normalization steps intended to remove non-linguistic noise while leaving intact those elements that may carry contextual information. In particular, hyperlinks in both `http(s)://` and `www.` format are removed, as

they rarely contribute to the target classification task, and instead they frequently introduce irregular token sequences.

Crucially, emoji characters were not removed. Instead, they were converted into their textual description using the `emoji.demojize` function (e.g., `:` → `smiling face`). This guarantees that the model will continue to have access to important information conveyed through emojis, while still mapping them to a form that is consistently handled by the tokenizer. Since Transformer tokenizers are trained on large multilingual corpora containing emoji descriptions, this strategy allows emojis to contribute meaningfully to the contextual representation learned by the model.

No lemmatization, stopword removal, punctuation removal, or lowercase conversion was performed. Such operations, advantageous for sparse lexical models, would modify the statistics of the subwords on which the transformation models are based and eliminate the morphological information that XLM-R can exploit.

After these first cleaning steps, each message was tokenized using the `xlm-roberta-base` tokenizer from HuggingFace’s `AutoTokenizer`. The tokenizer applies Byte-Pair Encoding (BPE), splitting words into multilingual subword units that the pretrained model can interpret consistently across Italian, Spanish, and English. Each processed instance is transformed into the standard input structure required by the model, with sequences truncated or padded to a fixed maximum length to allow efficient batching during training.

This preprocessing pipeline yields a representation that preserves the expressive richness of social media text while ensuring compatibility with the multilingual embedding space encoded by XLM-R. By maintaining contextual signals such as emojis, punctuation, and informal markers, the model is better equipped to identify underlying patterns associated with reclamatory language, which may not be detectable in strongly normalized text.

3.3. Text Classification Models

Following the embedding representations described in the previous section, we evaluate two linear, supervised learning algorithms widely used in text classification: Logistic Regression and Support Vector Machines (SVMs). Both models are well-suited for the high-dimensional and sparse feature spaces produced by the embedding representations introduced in Section 2.2.

Logistic regression is a linear classification model that learns a decision boundary by assigning weights to the input features. In text classification, it offers several advantages: as we said it handles sparse and high-dimensional feature vectors efficiently, it remains relatively interpretable since its coefficients directly correspond to lexical features, and it consistently performs well as a baseline method for linearly separable problems. In particular, in our task, where the goal is to predict a binary Reappropriation label, Logistic Regression represents a reliable and computationally lightweight option.

Support Vector Machines are another standard choice for text classification in sparse feature spaces. The model aims to identify the hyperplane that maximizes the margin between classes, resulting in improved generalization in noisy or imbalanced datasets.

3.4. Training setup

To ensure a consistent and reproducible evaluation framework, all models developed in this project were trained under a unified training setup encompassing data splitting strategies, feature construction, hyperparameter selection, and optimization details. Given the multilingual and multi-task nature of the project, the setup was adapted to the specific characteristics of the two tasks, while maintaining shared methodological principles.

3.4.1. Data Split

For Task A, we rely on the multilingual dataset introduced in Section 1. Following a standard practice in supervised text classification, the corpus was divided into a training set (80%) and a validation set

(20%), using a stratified split to preserve the class distribution of the Reappropriation label. The split is performed with a fixed random seed (equal to 42) to ensure reproducibility across experiments.

We then used a slightly different split for Task B. Each monolingual dataset is partitioned into 60% training and 40% validation. This choice reflects the smaller size of the task-specific corpora and helps to ensure the availability of a sufficiently large validation set for the reliable comparison of models and the tuning of hyperparameters. As in Task A, the division is stratified and deterministic. This validation strategy was adopted to simulate the official evaluation scenario of the shared task, where systems are assessed on an unseen test set provided by the organizers. Using a fixed development split, therefore, ensures methodological consistency and reproducibility across experiments.

3.4.2. Feature Construction and Hyperparameter Choices

The training process builds directly on the embedding representations presented in Section 2.2. For Task A, we employ `CountVectorizer` and `TfidfVectorizer`, configuring them to model not only single words but also short contiguous word sequences of up to three terms. To maintain tractability, the vocabulary is restricted to the 10,000 most frequent features.

For Task B, the training setup builds on the enriched text field produced during preprocessing, which integrates the tweet with its available contextual profile information. To effectively support this input space, we evaluate two TF-IDF configurations: a unigram-bigram model with a maximum vocabulary of 20,000 features, designed to capture short lexical patterns, and a more compact unigram-only model, which often provides greater robustness in low-resource or highly sparse settings.

Hyperparameter tuning is carried out directly on the validation splits. In the multilingual setting, both Linear SVMs and Logistic Regression models are explored using balanced class weights to mitigate the strong label imbalance. Linear SVMs are trained with conservative regularization ($C = 0.001$), while Logistic Regression employs a slightly stronger penalty ($C = 1$) together with an increased iteration limit to ensure convergence (`max_iter = 500`).

3.4.3. Optimization and Validation

All models are implemented using `scikit-learn`, whose deterministic optimization routines are well suited to the sparse, high-dimensional feature spaces produced by our embedding representations. As a consequence, the training process is not organized around epochs or mini-batches. Instead, each optimization step operates on the entire feature matrix, and training terminates once the library's internal convergence criteria are satisfied. Logistic Regression models are optimized with the default solver, whereas Linear SVMs rely on the efficient hinge-loss implementation provided by `LinearSVC`.

Given the size of the multilingual feature matrices and the computational overhead of repeated training, we do not employ k-fold cross-validation. Rather, we adopt a single, fixed development split for all experiments. This split serves multiple purposes: it provides a consistent basis for model selection, supports hyperparameter tuning and threshold calibration, and enables general diagnostic checks (including early detection of overfitting). Throughout this process, performance is systematically assessed using the Macro F1 score, the official evaluation metric of the shared task.

Finally, in order to guarantee reproducibility, all components involving randomness (e.g., data splitting and model initialization) are controlled through a fixed `random_state = 42`. This ensures that results are consistent across runs.

3.4.4. Setup for XLM-R

Unlike the linear models trained on sparse lexical features, the XLM-RoBERTa model required a distinct training setup.

Fine-tuning was performed using HuggingFace's Trainer API with the AdamW optimizer and mini-batch gradient descent. Unlike the `scikit-learn` models, which operate on the full feature matrix in a single optimization pass, the Transformer is trained for multiple epochs, updating parameters iteratively

over mini-batches. We did not manually set a custom learning rate schedule or warm-up steps. Instead, we used the default values provided by the HuggingFace Trainer.

To address the strong imbalance in the Reappropriation label, we employed a class-weighted loss function. This was implemented through a custom weighted trainer class that injects class weights into the cross-entropy objective during training.

Then, as in Task A, 20% of the multilingual dataset was held out as a validation set using a stratified split. The trainer evaluated the model at the end of each epoch, selecting the checkpoint with the highest Macro F1 score. This ensures comparability with the linear models, which also rely on Macro F1. Training a Transformer on a GPU does not produce exactly the same results every time. However, to make the results as reproducible as possible, the code still sets random seeds wherever the framework allows it.

Once the best hyperparameters were identified (selected empirically using the validation split described above), the model was additionally trained on the full training partition to maximize data usage before generating final predictions for evaluation.

4. Evaluation

This section describes the evaluation protocol adopted in our experiments, with particular attention to the metric required, the Macro F1-score, and to the preliminary validation procedure conducted before the official development labels were released. The goal is to detail how model performance was assessed in a controlled and reproducible manner.

4.1. Macro F1-score

The official evaluation metric for the shared task is the Macro F1-score, defined as the unweighted average of the F1-scores computed for each class. Unlike accuracy, which can be misleading when classes are imbalanced, the Macro F1 gives equal importance to both classes, regardless of their frequency. Why is Macro F1 appropriate for our imbalanced data? Our dataset exhibits a strong imbalance: Label 0 is substantially more frequent, while Label 1 is harder to identify. In such scenarios Macro F1 prevents this by penalizing models that perform poorly on the minority class. Thus, this metric directly reflects the system’s ability to detect true instances of reappropriation.

4.2. Preliminary Evaluation on a Training Split

Before the official validation set was released, model development relied on an internal validation split extracted from the training set (80/20). The main purposes of this internal split were to enable early performance estimation, and to guide hyperparameter selection.

The split preserved the original label distribution: the training portion contained 2,390 samples, while the validation portion contained 598 samples, with an identical imbalance of $\approx 85.7\%$ class 0 vs. $\approx 14.3\%$ class 1 in both partitions. This confirms that stratification was successful and that the validation subset is representative of the real data distribution.

4.3. Preliminary Results for Task A

4.3.1. Sparse Models

Table 1 reports the Macro F1-scores on the validation split for sparse lexical models using Bag-of-Words (BoW) and TF-IDF representations.

Among sparse models, **Logistic Regression** clearly outperforms LinearSVC, with **Bag-of-Words** yielding the strongest performance.

Given its strong performance, Logistic Regression with Bag-of-Words was further tuned by adjusting the inverse regularization parameter C . The best result was obtained with $C = 5.0$, reaching a Macro F1-score of 0.760.

Table 1

Preliminary results of sparse models for Task A (validation split).

Model	Representation	Macro F1
Logistic Regression	Bag-of-Words	0.758
Logistic Regression	TF-IDF	0.742
Linear SVM	Bag-of-Words	0.699
Linear SVM	TF-IDF	0.691

4.3.2. XLM-R

Finally, the multilingual Transformer XLM-R Base was fine-tuned on the same internal split. Table 2 reports the validation performance of XLM-R across training epochs.

Table 2

Preliminary results of the multilingual Transformer for Task A (validation split).

Epoch	Training Loss	Validation Loss	Macro F1
1	0.679	0.702	0.461
2	0.567	0.513	0.667
3	0.439	0.540	0.740

The model established a peak performance with a Macro F1 score of 0.7403 on the validation set at the conclusion of Epoch 3.

The training exhibited typical behavior for imbalanced classification: the Validation Loss reached its minimum at Epoch 2 (0.5134) and then slightly increased at Epoch 3 (0.5403), even as the target Macro F1 metric improved significantly. This slight divergence, coupled with a consistently dropping Training Loss (from 0.57 to 0.44), suggests an early onset of overfitting or a lack of perfect alignment between the cross-entropy loss function and the Macro F1 metric.

As expected, the contextual multilingual representation provided by XLM-R captures insights more effectively than lexical features.

4.4. Preliminary Results for Task B

Task B uses the same internal split (2,390/598 samples), but additionally allows the use of author metadata for Italian and Spanish.

4.4.1. Sparse Models with Metadata

The integration of metadata leads to consistent improvements over Task A, particularly for Logistic Regression. Table 3 reports the Macro F1-scores of sparse models on Task B.

Table 3

Preliminary results of sparse models for Task B.

Model	Representation	Macro F1
Linear SVM	Bag-of-Words	0.704
Linear SVM	TF-IDF	0.708
Logistic Regression	Bag-of-Words	0.789
Logistic Regression	TF-IDF	0.754

Logistic Regression with **Bag-of-Words** achieves the highest performance among sparse models, confirming the usefulness of metadata for improving minority-class recall.

4.4.2. XLM-R

XLM-R Base was also fine-tuned for Task B. The model achieves the following Macro F1-scores. Table 4 reports the Macro F1-scores obtained by XLM-R on Task B across epochs.

Table 4

Preliminary results of the multilingual Transformer for Task B.

Epoch	Training Loss	Validation Loss	Macro F1
1	0.723	0.722	0.452
2	0.654	0.491	0.734
3	0.482	0.520	0.766

The Macro F1 score improves substantially across every epoch, reaching a peak of 0.7659 in Epoch 3.

4.5. Summary

The internal train-validation split enabled early comparison of models and guided the selection of hyperparameters before training on the full dataset. Overall, the strongest sparse model was Logistic Regression with Bag-of-Words, which achieved a Macro F1 of 0.760 on Task A and 0.789 on Task B. However, the best overall performance across both tasks was obtained by the fine-tuned XLM-R Base model, which consistently outperformed all sparse lexical baselines and achieved the most balanced improvement across both majority and minority classes. Based on these results, the best-performing configurations for each task, the tuned Logistic Regression model, and the fine-tuned XLM-R, will be retrained on the complete training set and subsequently used to generate predictions on the official test data provided by the organizers.

5. Results

This section presents the experimental results obtained on the MultiPRIDE tasks, focusing on a comparative evaluation of the different modeling strategies. We first report a quantitative comparison across models and tasks, and then provide a detailed analysis of the best-performing systems, including qualitative observations and limitations.

5.1. Model Performance Comparison

Table 5 presents the Macro F1-scores obtained by the submitted models on Task A and Task B.

For Task A, which relies exclusively on the textual content of the message, both models achieve competitive performance. Logistic Regression with Bag-of-Words provides a strong baseline, confirming that lexical information alone captures a substantial portion of the signal associated with reappropriative language. However, XLM-RoBERTa achieves higher performance, demonstrating its ability to exploit contextual cues and cross-lingual semantic patterns that are not accessible to sparse representations.

For Task B, which incorporates author metadata through the bio field, performance improves for both models. The Logistic Regression model benefits from the additional contextual information, showing a clear increase in Macro F1-score compared to Task A. Nevertheless, XLM-RoBERTa remains the best-performing model overall, with a larger margin over the sparse baseline. This indicates that the Transformer is particularly effective at jointly modeling message content and author context.

Across both tasks, the results show a consistent trend: while Bag-of-Words combined with Logistic Regression constitutes a robust and efficient baseline, contextualized representations provide a measurable advantage, especially when additional metadata are available.

Table 5

Performance of the submitted models on Task A and Task B (Macro F1 and Accuracy).

Task	Model	Macro F1	Accuracy
Task A	Logistic Regression + BoW	0.758	0.875
	XLM-RoBERTa	0.740	0.895
Task B	Logistic Regression + BoW	0.789	0.893
	XLM-RoBERTa	0.766	0.860

5.2. Best-Performing Models

For both Task A and Task B, the best-performing system in terms of Macro F1 is Logistic Regression with Bag-of-Words.

In Task A, the sparse lexical model achieves slightly higher performance, confirming that surface-level lexical information captures a substantial portion of the signal associated with reappropriative language. However, XLM-RoBERTa remains competitive and demonstrates the ability to model contextual and semantic nuances beyond explicit lexical markers.

In Task B, Logistic Regression again achieves the highest Macro F1 score. Nevertheless, XLM-RoBERTa shows strong performance.

5.2.1. Qualitative Observations

Qualitative inspection indicates that both models perform reliably when reappropriation is explicitly marked through identity-affirming language. Errors tend to occur in highly ambiguous cases or in very short messages, where intent is difficult to infer regardless of the modeling approach.

5.2.2. Limitations

Despite their effectiveness, both submitted models face limitations. Logistic Regression remains constrained by its reliance on surface-level lexical features and limited contextual interaction. XLM-RoBERTa, while more expressive, requires greater computational resources and still struggles with subtle pragmatic phenomena such as irony and sarcasm. Additionally, the absence of author metadata in the English subset restricts the potential gains of Task B for that language.

Overall, the submitted systems illustrate the trade-off between efficiency and expressive power, while confirming the importance of contextual and metadata-aware modeling for the detection of reappropriative language.

The examples reported in Figure 8 highlight typical failure cases of both our models in the detection of reappropriative language. Most errors arise from lexical ambiguity and pragmatic ambiguity. In several cases, the model fails to distinguish between reclaimed in-group usage and genuinely derogatory usage, due to the semantic ambiguity, as the same lexical item can appear either as a slur or in a neutral historical or literal meaning.

5.3. Official Evaluation Results

This subsection reports the final evaluation results computed by the MultiPRIDE organisers on the official labeled test sets. Different from our internal validation experiments, where models were trained on a single multilingual dataset obtained by merging Italian, Spanish, and English data, the official evaluation was conducted using *language-specific training setups*. That is, separate models were trained and evaluated for each language, following the shared task guidelines.

Tables 6 and 7 report the Macro F1-scores obtained by the submitted systems for Task A and Task B, respectively.

id	text	lang
en_1506	If someone calls a straight person a faggot, does it cease to be a slur against queer people?	1
en_802	Upvote for fucking hilariousness and the fact that at least faggot correctly genders me, or downvote for bigotry? I can't make up my mind! Wat do?!	1
en_1789	Weekend is here. What an amazing week this has been. Let's use this extended weekend to celebrate our successes my fellow queer folk.	0
en_264	I think using a term in its original context is fine. * What a queer set of circumstances * What a queer set of circumstances * The young woman had a bounce in her step because she was feeling gay * The brit lit the fag and and took a drag after a long day's work * The young page collected a faggot of brushwood for the fire.	0
en_332	there's video of her calling a gay flight attendant a faggot after he allegedly hit her in the face, even though he wasn't the one who elbowed her, it was another passenger.	0

Figure 8: Qualitative analysis of misclassified examples.

Task A. For Task A, which relies exclusively on message text, XLM-RoBERTa consistently outperforms the Logistic Regression baseline in English, Spanish, and Italian. The performance gap is particularly evident in English, where the Transformer-based model shows a clear advantage in capturing contextual and semantic information. In Italian, both models achieve strong results, with XLM-RoBERTa obtaining the highest Macro F1-score overall. These findings confirm that contextualized multilingual representations generalize well even when trained on language-specific data.

Task B. For Task B, which incorporates author metadata, results are reported only for Italian and Spanish, as biographical information is not available for English. In Spanish, the two models achieve comparable Macro F1-scores, with Logistic Regression showing slightly higher recall-driven performance. In Italian, however, XLM-RoBERTa substantially outperforms the sparse baseline, achieving the best overall result across both tasks and languages. This highlights the Transformer’s ability to effectively integrate message content with author-level contextual information.

Overall, the official evaluation confirms the trends observed in our internal experiments: sparse lexical models provide strong and stable baselines, while Transformer-based models achieve superior performance, particularly in settings where richer contextual information is available.

Table 6

Official EVALITA results for Task A (Macro F1). Models trained separately for each language.

Language	Model	Macro F1
English	Logistic Regression + BoW	0.529
	XLM-RoBERTa	0.550
Spanish	Logistic Regression + BoW	0.710
	XLM-RoBERTa	0.778
Italian	Logistic Regression + BoW	0.834
	XLM-RoBERTa	0.850

Table 7

Official EVALITA results for Task B (Macro F1). Models trained separately for each language.

Language	Model	Macro F1
Spanish	Logistic Regression + BoW	0.720
	XLM-RoBERTa	0.719
Italian	Logistic Regression + BoW	0.832
	XLM-RoBERTa	0.883

6. Conclusions and Future Work

In this work, we presented the systems developed by the Hate Busters team for the MultiPRIDE shared task at EVALITA 2026, addressing the automatic detection of reappropriative intent in multilingual social media content related to the LGBTQ+ community. We explored both sparse lexical models and contextualized Transformer-based approaches, evaluating their effectiveness across two tasks with increasing levels of contextual information.

The experimental results highlight several key findings. First, Logistic Regression with Bag-of-Words features proved to be a strong and reliable baseline across all settings. Despite its simplicity, this model achieved competitive Macro F1-scores, particularly in Italian and Spanish, confirming that surface lexical cues already encode a substantial portion of the information relevant to reappropriative language.

Second, the fine-tuned XLM-RoBERTa model consistently achieved the best overall performance across both Task A and Task B. The advantage of the Transformer-based approach was especially evident in the official EVALITA evaluation, where models were trained separately on language-specific datasets. In this setting, XLM-RoBERTa demonstrated strong generalization capabilities across English, Spanish, and Italian, confirming its ability to capture contextual, semantic, and pragmatic aspects of reappropriative language that go beyond surface-level lexical patterns.

The inclusion of author metadata in Task B further improved performance for both models. In particular, the official evaluation results show that biographical information plays a substantial role in clarifying reclaimed versus derogatory language, especially in Italian. This confirms that reappropriative intent is often linked to speaker identity and social positioning, which cannot be deduced from message text alone.

Despite these positive results, several limitations remain. From a data perspective, the dataset is strongly imbalanced, with reclaimed instances representing a minority class, which continues to pose challenges for recall. In addition, the absence of the bio field for English limits the potential effectiveness of Task B for that language and reduces cross-lingual comparability. From a methodological point of view, our internal experiments relied on a merged multilingual training setup, while the official evaluation employed language-specific training. While both settings demonstrate consistent trends, this discrepancy indicates the potential value of a more systematic comparison between multilingual and monolingual training approaches.

Future work can explore multiple directions. First, extending the availability of author metadata to English could significantly improve performance in Task B and enrich the model usage. Secondly, the employment of language-specific Transformer models, as opposed to a unified multilingual architecture (as is the case with XLM-R), may prove more effective in capturing the fine-grained linguistic and cultural particularities. Finally, addressing data imbalance through targeted data augmentation or cost-sensitive learning may further improve minority-class detection. In addition, future work may explore the integration of Large Language Models, which have demonstrated strong capabilities in modeling pragmatic inference and contextual reasoning. Their ability to leverage instruction tuning and few-shot learning could further improve the detection of subtle forms of reappropriative language.

Overall, this work confirms the importance of contextual and metadata-aware modeling for the detection of reappropriative language and provides a solid baseline and reference point for future research on intent-sensitive text classification in multilingual and socially grounded settings.

Code and Data Availability

For reproducibility purposes, the source code is publicly available at: <https://github.com/camillagentili/multipride-reappropriative-intent-hatebusters/>. To obtain access to the data, we invite the reader to refer to the guidelines published in [6].

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4 and Grammarly for grammar and spelling checks, paraphrasing, and rewording. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] M. Popa-Wyatt, Reclamation: Taking back control of words, *Grazer Philosophische Studien* 97 (2020) 159–176.
- [2] B. Cepollaro, D. L. de Sa, The successes of reclamation, *Synthese* 202 (2023) 205.
- [3] C. Bianchi, Slurs and appropriation: An echoic account, *Journal of Pragmatics* 66 (2014) 35–44.
- [4] M. G. Worthen, Queer identities in the 21st century: Reclamation and stigma, *Current Opinion in Psychology* 49 (2023) 101512.
- [5] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [6] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [7] V. Basile, M. Lai, M. Sanguinetti, Long-term social media data collection at the university of turin, in: *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, CEUR Workshop Proceedings, 2018.
- [8] J. Mata, E. Gualda, A dataset of spanish tweets on people and communities lgbtqi+ during the covid-19 pandemic 2020–2022, *Zenodo*, 2025. doi:10.5281/zenodo.14878434.
- [9] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: *Proceedings of the 16th ACM Web Science Conference (WebSci '24)*, 2024, pp. 241–249.
- [10] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 8440–8451.

UniBO-FICLIT at MultiPRIDE: Fine-Tuning an ELECTRA-Based Model for the Detection of Italian Reclaimed Slurs

Simone Casazza^{1,*}

¹University of Bologna, Bologna, Italy

Abstract

Linguistic reclamation, where targeted communities reappropriate slurs or identity-related terms for non derogatory, affiliative, or dissociative purposes, poses a distinctive challenge for hate speech detection in NLP. This report presents an encoder-based system for the EVALITA 2026 MultiPRIDE task on Italian tweets, employing an ELECTRA architecture fine tuned for binary sequence classification in two settings: text only and text plus limited author context. Results indicate that the system reliably recognizes tweets that do not contain reclaimed language, whereas detecting tweets that do contain reclamation remains difficult, even when author biographies are included. Error analysis highlights three recurrent drivers of misclassification: orthographic variation in slurs, explicit attributive uses by users, and ambiguous stance signals in the absence of clear context. These findings underscore the methodological limits of a strictly binary framing. The study motivates broader modeling perspectives alongside practical steps within current pipelines, including more robust preprocessing to mitigate orthographic noise and exploration of data augmentation and rebalancing strategies under skewed training distributions.

Keywords

ELECTRA, fine-tuning, linguistic reclamation

Warning: This paper contains examples of explicitly offensive content.

1. Introduction

Hate speech detection is a well-established task in natural language processing (NLP), yet comparatively little attention has been devoted to the closely related phenomenon of linguistic reclamation, the practice whereby targeted communities reappropriate slurs or identity-related terms for non-derogatory, affiliative, or dissociative purposes. From a computational perspective, this poses a distinctive challenge: the same lexical item may function either as an expression of hostility or as a marker of in-group solidarity, irony, or critical stance, depending on context, speaker identity, and audience uptake. In operational systems, this ambiguity can inflate false positives, which not only degrade model utility but risk further marginalizing LGBTQ+ users by erroneously flagging and suppressing their speech, thereby reproducing harms against a group that is already subject to social and institutional oppression.

This report investigates reclamation detection within the MultiPRIDE framework [1] at EVALITA 2026, focusing on Italian tweets that contain LGBTQ+-related terms. The approach adopts an encoder-based ELECTRA architecture fine-tuned for binary sequence classification, examining a text-only setting alongside a setting that incorporates limited author context. The analysis highlights two structural hurdles, class imbalance and the pragmatic complexity of reclamation, that jointly hinder reliable identification. Results indicate that recognizing tweets without reclaimed language is comparatively easier than identifying reclamation itself, underscoring the limits of a binary framing for a context-dependent, multi-dimensional phenomenon.

Beyond establishing a baseline through fine-tuning ELECTRA, the study maps error concentrations

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ simone.casazza@studio.unibo.it (S. Casazza)

🌐 <https://github.com/simocasaz> (S. Casazza)

🆔 00009-0003-2593-9030 (S. Casazza)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

around orthographic variation, attributive uses, and ambiguous stance cues. Taken together, the findings motivate a broader modeling perspective beyond strict binary outcomes, alongside practical steps, more robust preprocessing to handle orthographic noise and systematic exploration of data augmentation and rebalancing, to strengthen learning under skewed and limited data conditions.

2. Related work

Hate speech detection as a natural language processing (NLP) task has been widely addressed in the literature (Fortuna and Nunes 2018; Poletto et al. 2021) [2] [3], but little attention has been given to the phenomenon of linguistic reclamation. This phenomenon has long been studied by linguists and philosophers of language (Brontsema 2004; Hom 2008; Anderson and Lepore 2013; Bianchi 2014; Cepollaro and López de Sa 2023) [4] [5] [6] [7] [8] because of its close connection to the nature of slurs and, consequently, to the nature of meaning in language.

From an NLP perspective, recent work has begun to address this challenge. Zsisku et al. (2024) [9] highlight a pragmatic issue: reclaimed language used by targeted groups can be misclassified as hate speech by automatic detectors that are not trained to recognize reclamation. To mitigate this, the authors introduced the RHDS dataset and fine-tuned a RoBERTa model previously trained on hate speech detection. Their approach significantly reduced false positives.

For Italian, the only relevant study is by Cuccarini et al. (2024) [10], who adapted the HODI dataset (Nozza et al. 2023) [11] by filtering tweets containing slurs using 17 terms from the HurtLex lexicon (Bassignana et al. 2018) [12]. They employed Qwen, a multilingual decoder-based model pre-trained on Italian, in a zero-shot setting with four progressively specific prompts.

Generally speaking, fine-tuning pre-trained encoder models for text classification tasks, especially in hate speech detection, has proven effective. For instance, Lavergne et al. (2020) [13] developed the best-performing model in the EVALITA 2020 HaSpeeDe 2 task (Sanguinetti et al. 2020) [14], while Locatelli and Locatelli (2023) [15] achieved top results in the EVALITA 2023 HODI task by fine-tuning a transformer-based model. Both cases confirm the strong performance of encoder architectures in Italian hate speech detection.

3. Task description

The MultiPRIDE task at EVALITA 2026 addresses the challenge of detecting reclaimed language in the LGBTQ+ context, where slurs or identity-related terms are used with a reappropriative intent rather than as hate speech. Participants are asked to build systems capable of performing binary classification: given a sentence containing an LGBTQ+-related term, the system must determine whether the term is used in a reclamatory way or not.

The task is organized into two main settings. The first focuses exclusively on the textual content of the message, allowing participants to work in either a constrained mode, using only the provided training data, or an unconstrained mode, where additional annotated resources may be incorporated. The second setting extends the classification to include contextual information, specifically the author’s profile biography when available, alongside the text. Both settings include subtasks for different languages, with Italian, Spanish, and English covered in the textual setting, and Italian and Spanish in the contextual setting.

Evaluation is based on macro F1-score computed over the binary label. Baseline systems are provided by fine-tuning language-specific BERT models with weighted cross-entropy loss.

4. Dataset

The dataset for the Italian subtasks, sourced from the TWITA collection (Basile et al. 2018) [16], has been divided by the organizers into two distinct sets: a training set that constitutes 60% of the entire

Table 1
Training set label distribution

Label	Frequency
1 (reclamation)	879
0 (no reclamation)	207

dataset, and a test set that comprises the remaining 40% of the tweets. The training set includes a total of 1,086 tweets, of which 1,000 contain the authors’ biographies, providing valuable context for analysis.

It is important to note that the distribution of class labels within this dataset is unbalanced. Out of the total, 879 tweets have been classified as containing reclaimed slurs, while only 207 tweets are labeled as not containing reclaimed language (Table 1). This significant disparity highlights the challenges posed by such an unbalanced dataset, emphasizing the need for careful consideration in analysis and modeling approaches.

5. Methods

5.1. EDA and Pre-processing

An exploratory data analysis (EDA) was first conducted on the training set to identify patterns and inform preprocessing decisions. Since usernames and URLs were already anonymized, they were removed along with redundant white spaces. A frequency analysis of hashtags and emojis was performed to assess their potential contribution to classification. Differences in relative frequencies between classes suggested that both emojis and hashtags carry informative signals for distinguishing reclamatory from non-reclamatory uses. Consequently, these elements were retained in the text.

Further frequency analysis was conducted on words in general and on characteristic terms to evaluate their association with each class. For this purpose, the HurtLex lexicon (Bassignana et al. 2018) was employed. HurtLex is a multilingual lexicon of offensive and derogatory expressions derived from Tullio De Mauro’s *Le parole per ferire* and later expanded into a machine-readable resource for hate-speech research. The Italian seed lexicon groups over a thousand hate-related terms into 17 semantic categories, covering explicit slurs as well as stereotyped or contextually offensive descriptors (e.g., categories related to ethnicity, disability, sexuality, swear words, and crime). HurtLex was further enriched with part-of-speech information via MultiWordNet and with definitions and multilingual mappings via BabelNet, yielding a cross-lingual lexicon available in more than 50 languages. In this work, only the Italian terms belonging to the homophobic category were selected and regular expressions were used to generate feminine, masculine, and plural variants. Results indicate that most slurs appearing in reclamation tweets (e.g., *frocio*, *gay*) also occur in non-reclamation contexts. However, there are two notable exceptions, which appear most in reclamation tweets. The first one is *frocio*, which, as noted by Nossem (2019) [17] and Cuccarini et al. (2024), represents a localized adaptation of the English term queer. Such localized forms are often created through reappropriation and semantic redefinition, making *frocio* highly indicative of reclamation and rarely used in a non-reclaimed sense. The second one is the orthographic variation *forci*, which is a community-specific marker of pride for the LGBTQ+ community.

Following the exploratory data analysis, the original training set was partitioned into training and validation subsets in an 85:15 proportion. Then, tokenization was performed using the tokenizer associated with the selected ELECTRA model, implemented via Hugging Face’s `AutoTokenizer` class¹ to ensure compatibility with the pre-trained architecture. The column containing authors’ biographies was preserved only for subtask B1.

¹https://huggingface.co/docs/transformers/v4.14.1/en/model_doc/auto#transformers.AutoTokenizer

Table 2
Hyperparameter search space

Subtask	Learning rate	Batch size	Weight Decay	Warmup steps	Accumulation steps
A1	[1e-5, 5e-5]	{16, 32}	[0.01, 0.1]	[0, 500]	None
B1	[1e-5, 5e-5]	{8}	[0.01, 0.1]	[0, 500]	{1, 2, 4}

5.1.1. Tweets and bios concatenation

To utilize both tweets and user bios as input for fine-tuning in subtask B1, the two fields are concatenated during tokenization. The tweet is treated as the first sequence, and the bio is treated as the second sequence, with the tokenizer encoding them in the format: [CLS] tweet tokens [SEP] bio tokens [SEP]. This structure allows the model to process both sequences simultaneously, leveraging the contextual information from both the tweet and the bio for classification tasks.

To handle the model’s maximum input length, the tokenizer is configured to truncate only the bio if the combined length exceeds the limit, ensuring the tweet remains intact. This approach enables the model to learn the relationship between the two inputs and use them effectively for prediction.

5.2. Architecture and hardware

The system is built on an encoder-only transformer architecture, specifically an Italian variant of ELECTRA². ELECTRA introduces a novel pre-training objective called Replaced Token Detection (RTD), which differs from BERT’s Masked Language Modeling by training a discriminator to identify whether each token in the input is original or replaced by a generator (Clark et al. 2020). This approach provides training signals for every token rather than only the masked subset, making ELECTRA significantly more sample-efficient and enabling it to achieve better performance with fewer parameters compared to BERT-based models of similar size. Empirical evaluations confirm that ELECTRA often outperforms BERT under equivalent computational budgets, particularly for smaller models, while maintaining strong contextual representations (Clark et al. 2020).

For this project, the Italian ELECTRA model was fine-tuned for binary sequence classification. The architecture consists of the pre-trained ELECTRA encoder and a classification head implemented via Hugging Face’s `AutoModelForSequenceClassification`³. This head includes a dropout layer followed by a linear layer that outputs logits for two classes; softmax is applied only at inference to compute probabilities. Fine-tuning was performed using the Hugging Face Transformers library, leveraging the `Trainer` API⁴ for streamlined training and evaluation. Hyperparameter optimization was integrated through Optuna⁵, targeting learning rate, batch size, weight decay, and warmup steps, with macro F1-score as the optimization objective.

All the preprocessing and fine-tuning steps were performed with a local laptop Nvidia RTX 3060 GPU.

5.3. Hyperparameters tuning

For both subtasks, hyperparameter tuning was carried out using the Hugging Face Trainer API in combination with Optuna for automated optimization. The search spaces employed are reported in Table 2. Using identical search spaces for the two subtasks led to out-of-memory errors in subtask B1, primarily due to longer input sequences resulting from the concatenation of tweets and user bios. To mitigate this issue, the batch size search space for subtask B1 was reduced, and gradient accumulation steps were introduced as an additional parameter to be tuned. Furthermore, while 50 trials were executed for subtask A1, only 30 trials were performed for subtask B1. Each run was set to a maximum of 10

Table 3
Hyperparameters from best runs

Subtask	Learning rate	Batch size	Weight Decay	Warmup steps	Accumulation steps
A1	2.35e-5	32	0.078	1	None
B1	2.06e-5	8	0.014	291	4

Table 4
Test results

Subtask	Macro F1	Macro Precision	Macro Recall
A1	0.8706	0.9150	0.8395
B1	0.8643	0.8938	0.8416

epochs, with early stopping applied to terminate unpromising runs.

After completing the search for each subtask, the hyperparameters from the best-performing trial (Table 3) were used to train a new model from scratch. Hardware constraints were again considered during this phase: a safe batch size was fixed, and gradient accumulation was employed to replicate the effective batch size of the optimal configuration without exceeding memory limits.

6. Results and discussion

6.1. Results

Table 4 reports the performance metrics for both subtasks. Overall, the model attains strong results in the textual classification setting (A1), with slightly lower performance when contextual information (author biographies) is incorporated (B1). However, these outcomes should be interpreted with caution: the hyperparameter search spaces and optimization strategies differed across subtasks due to memory constraints, which limits the validity of direct comparisons between A1 and B1.

In B1, the inclusion of biographies yields a marginal increase in recall for tweets that contain reclaimed language, but this comes at the cost of precision, producing a negligible change in macro F1. These findings suggest that simple concatenation of contextual data does not substantially improve performance under the current setup. Moreover, the dataset’s limited size and pronounced class imbalance likely bias the classifier toward the majority class, further constraining generalization.

In short, while the architecture performs well in recognizing tweets that do not contain reclaimed language, reliably identifying tweets that contain reclamation language remains challenging, even with additional context. Future work should explore richer modeling of pragmatics and stance, alongside data augmentation and rebalancing strategies, to improve sensitivity to reclamatory language.

The leaderboard for Subtask A1 (Table 5) shows that the UniBO-FICLIT system ranks just below the official baseline, which is based on a BERT-style architecture fine-tuned with a weighted cross-entropy objective. This outcome aligns with earlier observations about the significant class imbalance in the dataset: because reclaimed tweets form only a small portion of the training data, optimizing with an unweighted loss tends to favor the majority class. Consequently, the baseline’s superior performance appears to stem more from its training strategy than from inherent architectural differences. It is reasonable to assume that integrating class weights into the loss function of the system presented in this work would have improved the system’s ability to identify reclaimed instances and potentially allowed it to reach the baseline’s performance. In this sense, the A1 results reinforce one of the central methodological limitations highlighted in the thesis.

²<https://huggingface.co/dbmdz/electra-base-italian-xxl-cased-discriminator>

³https://huggingface.co/docs/transformers/v4.14.1/en/model_doc/auto#transformers.AutoModel

⁴https://huggingface.co/docs/transformers/main_classes/trainer

⁵https://huggingface.co/docs/transformers/main/en/hpo_train

Table 5
Participants results for Subtask A1

Team	Run	F1-score
Ghavidel-Rajabi	1	0.8981
MilaNLP	1	0.8959
Ghavidel-Rajabi	2	0.8909
SaFe Tweets	1	0.8895
LlaNa	1	0.8835
GRUPPETTOZZO	1	0.8834
Challenger	1	0.8816
HateltOff	1	0.8809
GRUPPETTOZZO	2	0.8735
baseline	1	0.8731
UniBO-FICLIT	1	0.8707
NamDang	2	0.8589
HateltOff	2	0.8584
NetGuardAI	1	0.8537
The Hate Busters	2	0.8503
I2C	2	0.8435
NamDang	1	0.8407
Kenji-Endo	1	0.8360
The Hate Busters	1	0.8345
MilaNLP	2	0.8306
KIT-TIP-NLP	2	0.8103
Avahi	1	0.7904
KIT-TIP-NLP	1	0.7659
I2C	1	0.6202

A different scenario emerges for Subtask B1 (Table 6). In this setting, the baseline achieves a very strong score, with only one system managing to surpass it. This suggests that the baseline constitutes a particularly robust point of comparison, and that the contextual configuration of B1 poses additional modeling challenges. The fact that nearly all participating systems fall short of the baseline indicates that well-regularized encoder models remain difficult to outperform when dealing with contextual inputs such as user biographies, especially under the constraints of this shared task.

6.2. Error analysis

Systematic misclassifications can be grouped into three recurrent phenomena: (i) orthographic variation in slurs, (ii) echoic attribution through quotation marks or reported speech, and (iii) the absence or ambiguity of contextual stance cues such as author metadata.

Orthographic variation. Several false negatives involve orthographic variations of slurs (e.g., *forcio* for *frocio*), which degrade the lexical signal on which the model relies. Subword splits become rare, embeddings are poorly estimated, and the distinction between reclaimed and non-reclaimed usage is obscured. These cases highlight the need for robust preprocessing strategies—such as character-level augmentation or fuzzy matching against a curated slur lexicon—and model components less sensitive to orthographic noise.

Echoic attribution. A second cluster includes tweets where slurs appear in quotation marks or are explicitly attributed to other speakers. In these cases, the author does not straightforwardly deploy the slur but uses it in an attributive way, representing an utterance or thought attributed to someone else (Bianchi 2014) [7]. More specifically, these are echoic uses, where the speaker signals an attitude toward the attributed content—often dissociative or mocking. While Bianchi argues that all reappropriation cases are echoic, quotation marks make this use more explicit. The model, however, seems to rely excessively on the presence of specific slur tokens and their direct use by the author, failing to capture

Table 6
Participants results for Subtask B1

Team	Run	F1-score
LlaNa	1	0.9021
baseline	1	0.8981
GRUPPETTOZZO	1	0.8979
The Hate Busters	2	0.8827
MilaNLP	1	0.8827
GRUPPETTOZZO	2	0.8681
UniBO-FICLIT	1	0.8644
MilaNLP	2	0.8641
AIWizards	1	0.8564
AIWizards	2	0.8549
HateItOff	2	0.8462
The Hate Busters	1	0.8324
HateItOff	1	0.8319
SaFe Tweets	1	0.8164
Avahi	1	0.8031
DataSummit	1	0.7734
Kenji-Endo	1	0.7489
NetGuardAI	1	0.7405

the echoic dimension of the reclamation phenomenon.

Contextual stance cues. Some misclassifications hinge on missing or ambiguous author metadata. For instance, tweets such as “*Avete paura di non potere più picchiare e insultare i ‘frocì’???*” are difficult to classify without knowing whether the author belongs to the target group. Although reclamation can also be performed by out-group speakers (Bianchi 2014; Cepollaro and López de Sa 2023) [8], the lack of clear contextual signals complicates interpretation. This difficulty also explains certain false positives, where the author’s stance is ambiguous even for human annotators. For example, in “*Prima che le haters mi inizino a dire eH mA qUeStA è OmOfObla, volevo comunicarvi che frocia lo sono pure io #circozzi*”, the slur is self-directed in a non-derogatory way, yet the initial framing suggests prior accusations of homophobia, making the overall intent unclear.

Additional false positives occur when the system over-relies on specific slurs (e.g., *frocio*) as indicators of reclamation, even in contexts lacking any reclaiming attitude, or when ambiguity arises from terms that can function as slurs in some contexts but not others. These cases underscore the complexity of reclamation and the limitations of purely lexical approaches, suggesting the need for models that integrate pragmatic and contextual reasoning.

The patterns observed collectively suggest that reclaimed language is not well captured by a binary decision boundary. As suggested by Bianchi (2014) and Cepollaro and López de Sa (2023) reclamation depends on many aspects (speaker identity, audience uptake, speaker attitude), many of which are gradient, context-relative, and multi-dimensional rather than strictly present/absent. Felicity and accomplishment conditions vary across different settings (Cepollaro and López de Sa 2023; Brontsema 2004) [4]. This complexity argues for modeling approaches beyond binary classification. Future datasets should reflect these dimensions to better align annotations with the phenomenon’s structure.

7. Conclusion

This work presented an encoder-based approach to the MultiPRIDE task at EVALITA 2026, tackling the detection of reclaimed language in tweets containing LGBTQ+-related slurs. Fine-tuning ELECTRA yielded competitive performance, but persistent challenges remain—most notably under severe class imbalance and limited training data.

While the architecture performs well in recognizing tweets that do not contain reclaimed language,

reliably identifying tweets that contain reclamation language is difficult, even when author biographies are included. The marginal changes observed in the contextual setting suggest that simple concatenation is insufficient to capture reclamation. Moreover, differences in hyperparameter search spaces across subtasks caution against direct comparisons.

The error analysis highlights three main sources of misclassification: orthographic variation, slurs' attribution, and ambiguous stance cues. Some of these findings suggest that reclamation is too complex to be faithfully modeled as a binary task. The phenomenon is inherently multi-dimensional and context-dependent, often requiring distinctions among stance, target, intent, felicity and accomplishment.

Overall, progress will likely depend on broadening the modeling perspective beyond binary outcomes, while concurrently strengthening the empirical foundation: more resilient preprocessing to handle orthographic variation and systematic exploration of data augmentation and rebalancing to address limited and skewed training distributions.

Acknowledgments

The author thanks Professor Fabio Tamburini for his helpful advice during the development of this project.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5.2 in order to: Grammar and spelling check. After using the tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] C. Ferrando, L. Draetta, M. Madeddu, M. Sosto, V. Patti, P. Rosso, C. Bosco, J. Mata, E. Gualda, Multipride at evalita 2026: Overview of the multilingual automatic detection of slur reclamation in the lgbtq+ context task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] P. Fortuna, S. Nunes, A survey on automatic detection of hate speech in text, *ACM Comput. Surv.* 51 (2018). URL: <https://doi.org/10.1145/3232676>. doi:10.1145/3232676.
- [3] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, V. Patti, Resources and benchmark corpora for hate speech detection: a systematic review, *Language Resources and Evaluation* 55 (2021) 477–523. URL: <https://doi.org/10.1007/s10579-020-09502-8>. doi:10.1007/s10579-020-09502-8.
- [4] R. Brontsema, A queer revolution: Reconceptualizing the debate over linguistic reclamation, *Colorado Research in Linguistics* 17 (2014) 1–17. URL: <https://www.sciencedirect.com/science/article/pii/S0378216614000526>. doi:10.1016/j.pragma.2014.02.009.
- [5] C. Hom, The semantics of racial epithets, *Journal of Philosophy* 105 (2008) 416–440. doi:10.5840/jphil2008105834.
- [6] L. Anderson, E. Lepore, Slurring words, *Noûs* 47 (2013) 25–48. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-0068.2010.00820.x>. doi:10.1111/j.1468-0068.2010.00820.x.
- [7] C. Bianchi, Slurs and appropriation: An echoic account, *Journal of Pragmatics* 66 (2014) 35–44. URL: <https://www.sciencedirect.com/science/article/pii/S0378216614000526>. doi:10.1016/j.pragma.2014.02.009.
- [8] B. Cepollaro, D. López de Sa, The successes of reclamation, *Synthese* 202 (2023). URL: <https://doi.org/10.1007/s11229-023-04393-y>. doi:10.1007/s11229-023-04393-y.
- [9] E. Zsisku, A. Zubiaga, H. Dubossarsky, Hate speech detection and reclaimed language: Mitigating false positives and compounded discrimination, in: Proceedings of the 16th ACM Web Science

- Conference, WEBSCI '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 241–249. URL: <https://doi.org/10.1145/3614419.3644025>. doi:10.1145/3614419.3644025.
- [10] M. Cuccarini, L. Draetta, C. Ferrando, L. James, V. Patti, Reclaim project: Exploring italian slurs reappropriation with large language models, in: F. Dell’Orletta, A. Lenci, M. Simonetta, S. Rachele (Eds.), *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, volume 3878 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2024, pp. 1–8. URL: https://ceur-ws.org/Vol-3878/39_main_long.pdf.
- [11] D. Nozza, A. T. Cignarella, G. Damo, T. Caselli, V. Patti, Hodi at evalita 2023: Overview of the first shared task on homotransphobia detection in italian, in: M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi (Eds.), *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2023. URL: <https://ceur-ws.org/Vol-3473/paper26.pdf>.
- [12] E. Bassignana, V. Basile, V. Patti, Hurtlex: A multilingual lexicon of words to hurt, in: E. Cabrio, A. Mazzei, F. Tamburini (Eds.), *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2018. URL: <https://ceur-ws.org/Vol-2253/paper49.pdf>.
- [13] E. Lavergne, S. Rajkumar, G. Kovács, K. Murphy, Thenorth @ haspeede 2: Bert-based language model fine-tuning for italian hate speech detection, in: V. Basile, D. Croce, M. Di Maro, L. C. Passaro (Eds.), *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, volume 2765 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2020. URL: <https://ceur-ws.org/Vol-2765/paper135.pdf>.
- [14] M. Sanguinetti, G. Comandini, E. di Nuovo, S. Frenda, M. Stranisci, C. Bosco, T. Caselli, V. Patti, I. Russo, Haspeede 2 @ evalita2020: Overview of the evalita 2020 hate speech detection task, in: V. Basile, D. Croce, M. Di Maro, L. C. Passaro (Eds.), *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, volume 2765 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2020. URL: <https://ceur-ws.org/Vol-2765/paper135.pdf>.
- [15] D. Locatelli, L. Locatelli, Lcts at hodi: Homotransphobic speech detection on italian tweets, in: M. Lai, S. Menini, M. Polignano, V. Russo, R. Sprugnoli, G. Venturi (Eds.), *Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023)*, volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2023. URL: <https://ceur-ws.org/Vol-3473/paper30.pdf>.
- [16] V. Basile, M. Lai, M. Sanguinetti, Long-term social media data collection at the university of turin, in: E. Cabrio, A. Mazzei, F. Tamburini (Eds.), *Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018)*, volume 2253 of *CEUR Workshop Proceedings*, CEUR-WS.org, Aachen, 2018. URL: <https://ceur-ws.org/Vol-2253/paper48.pdf>.
- [17] E. Nossem, Queer, frocia, femminielle, ricchione et al.- localizing 'queer' in the italian context, *GSI: Gender, Sexuality, Italy* 6 (2019) 1–27. URL: <http://www.gendersexualityitaly.com/?p=2865>.

A. Online Resources

The source code of the project is available on github:

- <https://github.com/simocasaz/evalita2026-multipride-UniBO-FICLIT>,

Cruciverb-IT at EVALITA 2026: Overview of the Crossword Solving in Italian Task

Cristiano Ciaccio^{1,2}, Gabriele Sarti^{3†}, Alessio Miaschi², Felice Dell’Orletta² and Malvina Nissim⁴

¹Department of Computer Science, University of Pisa, Italy

²Institute for Computational Linguistics "A. Zampolli" (CNR-ILC) - ItaliaNLP Lab, Pisa, Italy

³Khoury College of Computer Sciences, Northeastern University, USA

⁴Center for Language and Cognition (CLCG), University of Groningen, The Netherlands

Abstract

Cruciverb-IT is the first shared task on Italian crossword solving, held at EVALITA 2026. The task comprises two subtasks: (1) answering individual crossword clues given the expected answer length, and (2) autonomously solving complete crossword grids of varying sizes. We release a dataset of approximately 410,000 Italian clue-answer pairs along with automatically generated crossword grids ranging from size 5×5 to 13×13. Five teams participated in the evaluation, submitting a total of 17 system runs. The best-performing system on Subtask 1 achieved 69% accuracy at rank 1 and 0.72 MRR using a retrieval-augmented LLM approach, while the top system on Subtask 2 reached an average character accuracy of 92%, fully solving 34% of grids by means of a fine-tuned encoder-decoder model paired with a constraint-driven depth first search and ranking heuristics. Results show that while modern approaches achieve strong performance on individual clues and smaller grids, solving larger crosswords remains an open problem, with full match performance decreasing rapidly for grids larger than 5×5.

Keywords

NLP, Crossword Solving, Evaluation, Language Models, Italian, Shared Task

1. Introduction and Background

Historically, language games have been an important testbed for creating and studying complex decision-making programs, largely due to a fundamental property: no fixed set of rules will be sufficient to define the overall gameplay. Given the involvement of natural language, in which the interpretation of meaning play a crucial role, judgment is needed not only to produce a solution but even to interpret the rules themselves and, since natural language can be used to describe the full range of human experiences [1], language games are inherently inconsistent with the closed-world assumption [2], according to which anything not explicitly defined is assumed not to hold.

Consequently, prior work has characterized language games such as crossword puzzles as AI-complete problems [3], as solving them requires human-level knowledge and natural language understanding capabilities. For these reasons, language games are emerging as valuable testbeds for evaluating and enhancing the reasoning abilities of Language Models (LMs).

Among language games, crossword puzzles represent a particularly challenging and multifaceted task that requires not only linguistic competence but also cultural knowledge, lateral thinking, and the ability to interpret ambiguous or polysemous clues [4, 5, 6, 7]. As a result, solving crosswords involves complex semantic and pragmatic reasoning, making this setting ideal for testing models’ deeper language understanding capabilities beyond surface-level aspects.

Before the advent of modern LMs, most approaches to crossword solving and clue answering relied on retrieval-based methods and shallow lexical and semantic features [3, 8]. For example, Barlacchi et al.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

[†]Work done at the University of Groningen, The Netherlands.

✉ cristiano.ciaccio@ilc.cnr.it (C. Ciaccio); g.sarti@northeastern.edu (G. Sarti); alessio.miaschi@ilc.cnr.it (A. Miaschi); felice.dellorletta@ilc.cnr.it (F. Dell’Orletta); m.nissim@rug.nl (M. Nissim)

🆔 0009-0001-6113-4761 (C. Ciaccio); 0000-0001-8715-2987 (G. Sarti); 0000-0002-0736-5411 (A. Miaschi); 0000-0003-3454-9387 (F. Dell’Orletta); 0000-0001-5289-0971 (M. Nissim)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

[9] proposed a system that exploited lexical resources and similarity metrics to match clues to candidate answers in Italian, while the SACRY system [10] incorporated syntactic information and ranking strategies to improve clue-answer matching. However, these systems typically struggle with clues that require deeper interpretative reasoning, such as wordplay, anagrams, or polysemous expressions. Consider, for instance, the clue “Producono con procedimenti lenti”, where “lenti” can mean both “slow” and “lenses” in Italian;¹ a viable answer could be *ottici* (opticians), illustrating the type of ambiguity traditional systems often fail to resolve.

Despite the impressive advancements in Large Language Models (LLMs), their performance on language games such as crosswords remains limited, especially in morphologically rich and less-resourced languages like Italian [11, 12, 13]. Existing LMs and retrieval-based systems still fall short when faced with clues requiring subtle reasoning or cultural grounding.

Building on this line of research, the Cruciverb-IT task organized at EVALITA 2026 [14] represents the first shared task specifically dedicated to crossword solving. The initiative was designed to encourage research in this direction by providing a challenging testbed for developing and evaluating systems on crossword puzzle solving.

2. Definition of the Task

The Cruciverb-IT shared task is organized into two subtasks:

Subtask 1: Clue Answering. The first task consists of answering clues extracted from Italian crosswords. Specifically, the task is formatted as a question-answering problem: participants are presented with a set of clues $C = \{c_1, c_2, \dots, c_n\}$ and are asked to build a system that for a given clue c_i is able to produce one or multiple candidate solutions $\hat{S} = \{\hat{s}_1, \hat{s}_2, \dots, \hat{s}_m\}$, possibly containing the correct answer s_i . To simulate a more realistic crossword solving scenario and to further guide the systems towards the correct answer space, each clue c_i is paired with the character length of the target answer s_i . For example: given the clue and the target character length *Sono un fiore di straordinaria bellezza*, 4, the systems should produce a list of one or more candidates, i.e. [*iris*, ***rosa***, *rose*, *yuzu*, *fior*, ...] eventually containing the correct answer ***rosa***.

Subtask 2: Grid Solving. The second task consists of autonomously solving Italian crossword grids. The participants are presented with a set of empty crossword grids $G = \{G_1, G_2, \dots, G_k\}$ where each grid G_i is paired with a list of clues, each one annotated with the (x, y) coordinates of the square where the corresponding solution starts in the grid and the direction, either down (*verticale*) or across (*orizzontale*). A crossword grid consists of a matrix G_i of size $\mathbb{R}^{n \times n}$ and each square is either blank or a black square. The developed systems should autonomously fill the grid with appropriate solutions, yielding a fully or partially filled crossword grid that ensures consistent overlap between the characters of crossing words and maximizes the number of correctly placed solutions.

2.1. Task Constraints

Participants were allowed to take part in either subtask or both. For both tasks, we enforced a set of constraints to ensure a fair comparison across systems and prevent training contamination. In particular, the use of external data sources that explicitly contain crossword clues or clue-answer pairs was strictly forbidden. All other types of external data and resources were permitted, including but not limited to dictionaries, encyclopedic resources (e.g., Wikipedia), lexical databases (e.g., WordNet), pre-trained or fine-tuned language models, and distributional representations. Participants were required to explicitly report all external data and resources used in developing their systems. This restriction was introduced to avoid trivial memorization effects and to prevent scenarios in which systems could

¹The phrase can be translated as “[they] produce with slow procedures”, or “[they] produce lenses with procedures”, with an unusual but acceptable constituent order in the latter.

exploit large collections of gold crossword data combined with highly engineered search strategies to achieve artificially high performance, potentially comparable to or even exceeding that of professional human solvers [15]². By disallowing crossword-specific external resources, we aimed to foster the development of models that genuinely address clue interpretation, lexical retrieval, and reasoning.

3. Dataset

For the proposed task, we relied on both the ItACW crossword dataset [16] and on a collection of additional clue-solution pairs found on the web. The final dataset, after duplicates are removed, contains approximately 410,000 clue-answer pairs, encompassing various types of puzzles, including wordplay, cryptic clues, named-entity initials, and fill-in-the-blank clues. For the first task, the dataset was divided into training (90%), validation (5%), and test (5%) sets, resulting in approximately 370,000 training examples, and 20,000 examples each for validation and testing. Splits were released as .csv files containing three columns: *clue*, *answer* and *answer_length*, with *answer* columns omitted in the test set.

For the second task, we automatically generated crossword grids by employing a constraint-driven, search-based construction algorithm designed to populate a predefined crossword layout with valid words from the list of answers contained in the aforementioned train, validation and test splits, respectively. Specifically, we first generated several empty and square matrices by placing black squares (with various proportions of the total number of squares) randomly, although ensuring symmetry in the layout, and, subsequently, we populated the grid with the aforementioned algorithm. Lastly, we collected the corresponding clues for each word in the grid, therefore obtaining several complete and plausible crosswords. We generated crosswords of different sizes in order to account for various levels of complexity: 5×5 , 7×7 , 9×9 , 11×11 and 13×13 with the number of black squares (as a percentage of the overall available squares) being, respectively, 15%, 16%, 22%, 27% and 27%. Specifically, each empty crossword grid is represented as a matrix, i.e. a list of lists, where each square is either blank (noted as a whitespace ' ') or a black square (noted as a dot '.'). On the other hand, given a grid, the corresponding clues are a list of dictionaries with keys *answer*, *clue*, *x*, *y*, *direction*, *length*, where the coordinates, (*x*, *y*, respectively, rows and columns) expresses where the corresponding solution starts in the grid, the length is the solution number of characters and the direction, noted as *D* or *A*, indicates if the solution should be placed either down (*D*) or across (*A*). As for the first task, we divided the dataset into training (500 grids), validation (50 grids) and test (50 grids) sets. More specifically, the grids were generated following a predefined distribution over grid sizes. The training set consists of 300 grids of size 5×5 , 150 of size 7×7 , 25 of size 9×9 , 15 of size 11×11 , and 10 of size 13×13 . Both the validation and test sets include 10 grids for each grid size (i.e., 10 grids of size 5×5 , 7×7 , 9×9 , 11×11 , and 13×13). Accordingly, participants were asked to return as output the completed crossword grids produced by their systems, represented in the same matrix-based format as the input empty grids, with blank squares filled with the predicted answers³.

4. Evaluation

The evaluation of the systems was conducted with specific metrics per task, as follows:

- Task-1: **Accuracy@1/10**, that is the accuracy in retrieving the correct solution word given the corresponding clue, considering the top 1 and 10 candidates produced by the system; **Mean Reciprocal Rank (MRR)**, that is the average of the reciprocal ranks of the first relevant item across all clues.
- Task-2: **% of correct characters (CharAcc, CA)**, that is the accuracy in inserting the correct characters in the correct slots; **% of correct words (WordAcc, WA)**, accuracy in inserting the

²<https://en.wikipedia.org/wiki/Dr.Fill>

³The dataset can be found at the following HuggingFace repository: <https://huggingface.co/datasets/cruciverb-it/evalita2026>

correct word in the correct slots; % of grids solved correctly (**FullMatch, FM**), the accuracy in solving the entire grid. Partially filled grids were evaluated by counting empty squares as errors.

Baselines For clues-answering, our baseline is obtained by approaching the task as an information retrieval problem: given a clue c_1 from the test set $C_{test} = \{c_1, \dots, c_n\}$, our system ranks the most similar clues by computing a similarity score between c_1 and each clue in the training set $C_{train} = \{c_1, \dots, c_m\}$. After selecting the top ten most similar clues, we extract the corresponding ten answers. The similarity scores between clues are estimated using the BM25 algorithm [17], a well-established ranking function in Information Retrieval. For solving crossword grids, our baseline is computed by combining the aforementioned ranker baseline with an additional module that optimizes for a solution by maximizing satisfied constraints while respecting the grid’s hard constraints. Specifically, by treating crossword puzzles as a weighted Max-SMT problem, as partially described in [18], the baseline optimizes for a solution by defining hard constraints (the grid structure) and soft constraints (candidate ranking preferences). Each clue corresponds to a disjunctive group of grid variables constrained to match candidate answers, combined conjunctively across the grid. The formulation uses the Z3 optimizer [19]⁴ with 10 candidates per clue.

5. Submitted Systems and Participants

Following a call for interest, 5 teams registered for the task and submitted their predictions, for a total of 17 runs (11 for subtask 1 and 6 for subtask 2). As shown in Table 1, three teams participated only in subtask 1, while two submitted runs for both tasks.

AC/DG [20] AC/DG adopts a retrieval-based framework that combines lexical, semantic, and hybrid reranking strategies. Given a clue and a target length, all retrieval methods operate under strict length constraints, restricting the search space to training instances whose solutions match the target length. The first component is a sparse lexical retriever based on BM25, designed to capture explicit term overlap and definitional clues. In parallel, a dense retrieval model, fine-tuned on Italian and based on a Sentence-BERT encoder, maps clues into a latent semantic space, allowing the system to retrieve morphologically and semantically related candidates even when lexical overlap is limited. The two retrieval streams are combined in a hybrid retrieve-and-rerank strategy, where the top candidates from both BM25 and dense retrieval are passed to an LLM (Qwen3 8B [21]) acting as a zero-shot judge. This model evaluates, reorders, and, when necessary, augments the candidate set by generating a fallback solution. The final output is obtained by selecting the highest-ranked answer from this reranked list, thereby balancing precision from lexical matching with semantic generalization and generative reasoning.

FFT-UniBa [22] FFT-UNIBA adopts a two-stage approach corresponding to the two subtasks. For Task 1, the authors fine-tune an encoder-decoder model based on IT5 [23], pre-trained on Italian texts, introducing length-aware special tokens to explicitly control answer generation. Each input is augmented with a pair of tokens marking the expected solution length, including a length-dependent end-of-sequence token. This design encourages the model to internalize length constraints during training, reducing generation errors caused by length mismatches without relying on post-hoc filtering. For Task 2, each crossword is formulated as a constraint satisfaction problem, where variables correspond to clue slots, and domains consist of ranked candidate answers generated by the Task 1 model. When necessary, the candidate sets are augmented with a small number of dictionary-based words matching the required length and letter pattern. Grid constraints enforce character consistency at word intersections. Crossword solving is performed via a depth-first backtracking search, initialized from single-word seed configurations and guided by model perplexity scores. To ensure tractable inference, the system

⁴We modified an open-source implementation: <https://github.com/pncnmp/Crossword-Solver>.

Team	Members	Affiliation	Task	Runs T1	Runs T2
AC/DG	4	Politecnico di Torino	1	3	-
FFT-UniBa	5	Università degli Studi di Bari Aldo Moro	1,2	4	4
MINDS	1	Politecnico di Torino	1	1	-
UNIBA	1	Università degli Studi di Bari Aldo Moro	1,2	2	2
UniTor	2	Reveal Srl; Università degli Studi di Roma Tor Vergata	1	1	-

Table 1

Teams participating in the EVALITA 2026 Cruciverb-IT shared task. For each team, we detail the number of team members, their affiliations, the sub-task(s) they participated in, and the number of submitted runs per subtask (T1 and T2).

enforces explicit limits on node expansions and dynamically adapts candidate cutoffs and search budgets based on grid size.

MINDS [24] MINDS frames crossword clue answering as a masked language modeling problem using an encoder-only architecture. Given a clue and the target word length in characters, the input is constructed by appending a templated sequence in which the answer is represented by a span of [MASK] tokens, together with an explicit indication of the expected length. An Italian BERT model is fine-tuned to reconstruct the masked span from the clue context, using standard masked language modeling with cross-entropy loss applied only to the answer positions. At inference time, since the number of subword tokens corresponding to the answer is unknown, the system queries the model with multiple hypothesized mask lengths. For each length, top- K predictions are extracted for each masked position and combined to form candidate answers, which are scored using the geometric mean of token probabilities. Candidates generated across different mask lengths are merged into a single ranked list, keeping the highest score for duplicates. Invalid candidates are pruned based on character length and symbol constraints, and the final output consists of the top-ranked valid answers. This strategy allows an encoder-only model to approximate variable-length generative behavior for crossword solving.

UNIBA [25] UNIBA addresses the crossword-solving task by mimicking a human-like incremental solving strategy based on partial solutions and cross-checking. The approach relies on an encoder-decoder transformer trained to generate answers conditioned not only on the clue, but also on a partially filled solution, when available. Training data are expanded by masking one or more characters in each gold answer, generating all possible partial solutions, and enriching the input with the number of missing characters and the expected answer length. The model is based on IT5-Large and is trained exclusively on the task-provided data, without external lexical resources. During inference, candidate answers are generated dynamically at each step based on the current grid state. The crossword filling task is instead performed using a beam search strategy that iteratively selects and expands the most promising partial grids. To improve candidate selection, the system employs a binary classifier trained on simulated crossword-solving trajectories, which scores candidate answers using grid-level, clue-level, and generation-based features. Candidates are ranked by classifier confidence and decoder scores, allowing the system to balance solution quality and search efficiency. The final submission excludes models using special tokens, which showed inferior validation performance.

UniTor [26] UniTor proposes a retrieval-grounded, LLM-based system that formulates crossword clue answering as a constrained ranking problem. Given a clue and target length, the system combines retrieval-augmented evidence with structured LLM prompting to generate and rank candidate solutions. In a first stage, UniTor retrieves length-compatible clue-solution pairs from a large indexed repository using lexical (i.e. BM25), neural (i.e. BGE-M3 [27]), or hybrid similarity, and injects them into the prompt as lightweight few-shot evidence. In a second stage, candidate generation and ranking are performed within a single LLM call through a structured two-phase prompt. The model is first instructed to explore a diverse set of plausible candidates, prioritizing recall and semantic coverage, and then to filter, normalize, and re-rank them under hard structural constraints such as exact length. This

Subtask 1				Subtask 2			
Systems	Acc@1	Acc@10	MRR	Systems	CA	WA	FM
UniTor	0.69	0.83	0.72	FFT-UniBa_c1000_2	0.92	0.85	0.34
FFT-UniBa_Constrained1	0.58	0.75	0.63	FFT-UniBa_c1000NODICT_2	0.92	0.85	0.32
MINDS	0.59	0.71	0.62	FFT-UniBa_c1000_1	0.93	0.84	0.28
FFT-UniBa_Constrained2	0.57	0.75	0.62	FFT-UniBa_c1000NODICT_1	0.91	0.82	0.22
FFT-UniBa_Unconstrained1	0.55	0.72	0.60	UNIBA RUN2	0.82	0.67	0.16
FFT-UniBa_Unconstrained2	0.54	0.73	0.60	UNIBA RUN1	0.82	0.66	0.16
AC/DG_Embeddings	0.51	0.73	0.57	Baseline	0.73	0.58	0.08
AC/DG_BM25	0.47	0.67	0.53				
AC/DG_Hybrid_Qwen3_Judge	0.46	0.69	0.52				
UNIBA RUN2	0.43	0.59	0.47				
Baseline	0.40	0.62	0.46				
UNIBA RUN1	0.36	0.54	0.41				

(a)

(b)

Table 2

Cruciverb-IT leaderboard. Subtask 1 (a) ranked by MRR; Subtask 2 (b) according to the three metrics: CharAcc (CA), WordAcc (WA) and FullMatch (FM). Subtask 2 results are ranked by FM.

explicit separation between exploration and selection is designed to improve ranking stability and constraint adherence while avoiding multiple LLM interactions. The system outputs a probability-ranked list of candidate answers, where scores represent relative confidence rather than calibrated probabilities. UniTor is evaluated across multiple instruction-tuned LLMs of varying scale to analyze the contributions of model capacity, retrieval grounding, and structured reasoning strategies to crossword-solving performance. The final submitted run is based on the GLM 4.6 model [28].

6. Results and Discussion

In the following, we report and discuss the results achieved by the participants with a further quantitative analysis highlighting relevant influencing factors across both tasks, the potential of an ensemble that leverages all participants predictions and similarities between systems predictions.

6.1. Subtask 1: Clue Answering

Table 2 (a) reports the leaderboard of systems participating in Subtask 1. First of all, we observe that most systems outperformed the baseline, indicating that the task cannot be reduced to a simple retrieval problem and that neural and hybrid approaches can capture deeper semantic and inferential patterns than a traditional BM25-based matching strategy. The retrieval-augmented LLM approach adopted by UniTor achieved the best performance, outperforming fine-tuned encoder-decoder models by a significant margin (+11% Acc@1 gain over the second-best system), leveraging a backbone LLM that is, parameter-wise, approximately 6000 times larger than the one used by FFT-UNIBA. This suggests that, while grounding generation with retrieved items provides a valuable contextual guidance for LLMs, the number of parameters remains a strong predictor of final performance, even in crossword clues. Despite this, the Acc@10 gap between UniTor and FFT-UNIBA, MINDS, AC/DG (ranging from 12% to 8%) with respect to the large gap in terms of free parameters, clearly showcases the standalone strength of traditional clues-retrieval approaches and the usage of small neural language models trained with a task-specific fine-tuning strategy, especially in terms of the efficiency/accuracy trade-off. To further support this, we replicate the asymmetrical dual-encoder approach proposed in Ciaccio et al. [13] on the official subtask 1 data, yielding an Acc@10 of 78%, further closing the gap (+5%) with the best system despite, again, a sensibly smaller parameter size⁵.

⁵We used the paraphrase-multilingual-mpnet-base-v2 in a dual-encoder setup with ≈ 556 M free parameters.

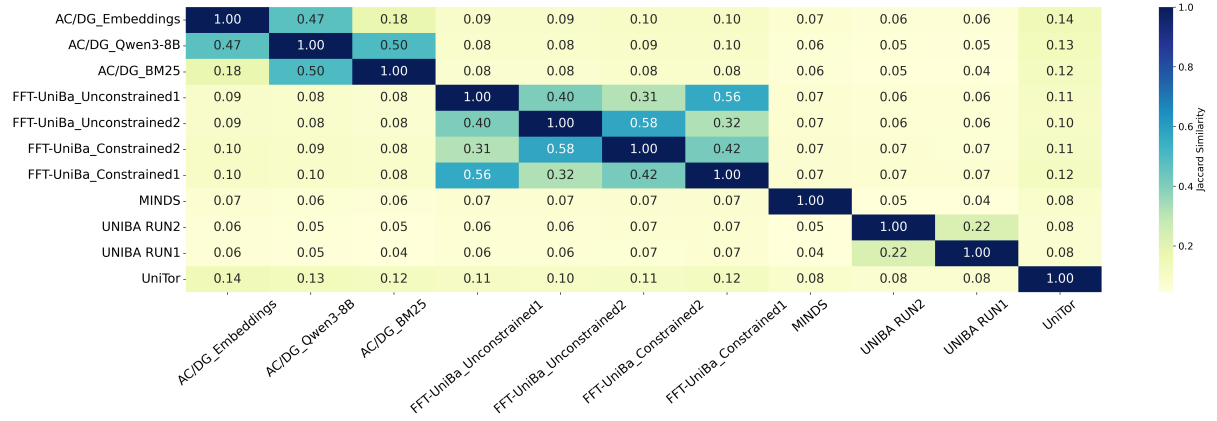


Figure 1: Average pairwise Jaccard similarity (Top10) between all systems sets predictions (a value of 1 indicates a perfect overlap).

System Similarity and Oracle Ensemble. To assess potential similarities between systems, we computed the average pairwise Jaccard similarity across all systems’ prediction sets. Specifically, given a test instance t_i and two systems f_o and f_p producing the candidates lists $\hat{S}_o$ and $\hat{S}_p$, the Jaccard similarity between $\hat{S}_o$ and $\hat{S}_p$ is obtained by $J_{t_i} = \frac{|\hat{S}_o \cap \hat{S}_p|}{|\hat{S}_o \cup \hat{S}_p|}$. By comparing all possible pairs of systems and averaging these values across all clues in the test set, as shown in Figure 1, we show that runs from the same team tend to cluster together and exhibit strong similarity, while there is almost no overlap across different teams. These results reveal that participants leveraged different approaches, yielding heterogeneous candidate lists. To further assess the impact of prediction diversity between systems, we built an oracle ensemble by taking the union of all systems’ top-k candidate sets ($k=1$ and $k=10$) for each clue, and counting a prediction as correct if any system included the gold answer. This approach resulted in upper-bound Acc@1 and Acc@10 of 85% and 94%, respectively. The marked improvements from the best system scores (Acc@1 +16%, Acc@10 +11%) highlight the diversity between systems’ predictions and the synergistic potential of combining the approaches proposed by the participants.

Influencing Factors. Several influencing factors were found with respect to the system’s predictions⁶. Specifically, accuracy scores tend to clearly decrease as the answer’s characters number increases, suggesting that longer answers are harder to predict while shorter ones are easier (see Figure 2, on the right) despite the presence of an initial drop for answers of length 2 (a set that usually includes wordplay, abbreviations, initials, etc.). Interestingly, while all systems follow the same trend, the UNIBA RUN2 is more resilient to this aspect, achieving competitive results for answers longer than 7 characters. Coherently, by inspecting the impact of the answers’ frequencies⁷, we found a strong positive correlation across all systems (see Figure 2, on the left). Moreover, we also report a negative correlation between Acc@10 and clues lengths – probably denoting longer clues that are harder to interpret – across all systems, ranging from -0.53 to -0.9, with the notable exception of UniTor showing no statistically significant correlation.

System Agreement and Lexical Exposure. We further analyzed the systems’ errors by inspecting the degree of agreement across participants. For each clue–answer instance in the test set, we computed the percentage of systems that correctly predicted the gold answer. We then investigated how this agreement relates to lexical exposure by distinguishing whether the gold answer appeared in the training data⁸. Our analysis reveals that lexical exposure has a pronounced effect primarily at higher agreement levels. Instances for which all systems correctly predicted the answer almost exclusively

⁶All reported correlations are Pearson coefficients with $p < 0.05$.

⁷Frequencies are computed on a 2021 Italian Wikipedia dump.

⁸The distribution of agreement levels with respect to training set coverage is reported in Appendix A.

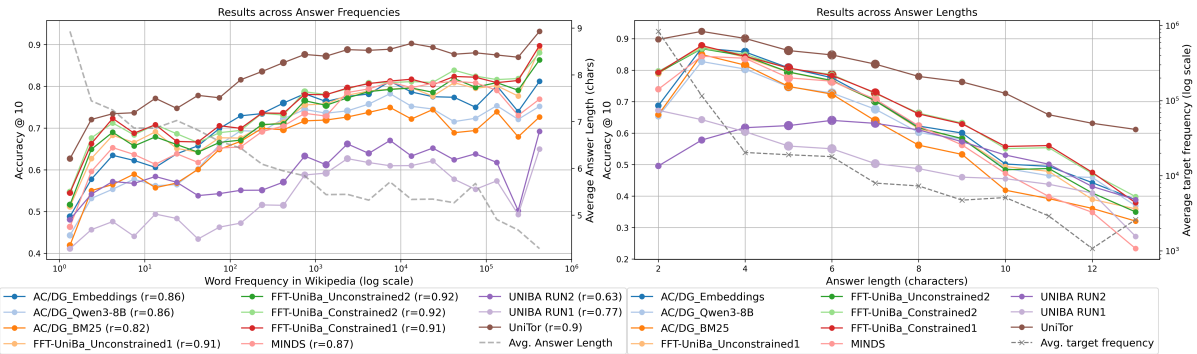


Figure 2: On the left (a), the plot shows the Acc@10 of each run across log frequency bins along with their respective Pearson (r) correlations; the gray dashed-line marks the average answer length per bin. On the right (b), the plot shows Acc@10 of each run across different answer lengths.

involve words that were observed during training (6236 and 2 instances, respectively), whereas unseen answers are almost absent in this subset. In contrast, for instances that were mistakenly predicted by (almost) all systems, the distribution between seen and unseen words is nearly balanced. These results suggest that, given the heavy training set dependence of the proposed approaches, the presence of an answer word in the training data has a substantial impact on the system’s ability to consistently retrieve it at test time, leading to strong agreement across participants. Conversely, in instances characterized by widespread errors, the presence or absence of the answer in the training set does not provide a clear advantage, indicating that lexical exposure alone is insufficient to overcome more challenging clues.

Qualitative Analysis of Shared Errors. Finally, we conducted a qualitative analysis of the errors shared by all systems, focusing on the subset of test instances that were consistently mistaken and therefore represent the most challenging cases. To this end, we applied a TF-IDF representation to the clues and performed unsupervised K-Means clustering to identify recurring patterns in these hard instances. Inspecting the results from a qualitative standpoint⁹, we noticed the presence of two clear macro-categories. A first group comprises clues that require access to specific cultural or encyclopedic knowledge (e.g., references to well-known public figures, films, or classical quotations), which are likely harder to solve. A second group consists of inherently ambiguous clues, for which a unique answer may not exist in isolation. Such clues are typically disambiguated only when embedded within a crossword grid (e.g., generic clues such as “*Città francese*” or “*Nome d’uomo*”¹⁰), a contextual constraint only available in Subtask 2.

6.2. Subtask 2: Grid Solving

Table 2 (b) reports the leaderboard of systems participating in Subtask 2. All systems achieved substantially higher results than the baseline thanks to a combination of stronger clue-answering experts and specifically tailored grid-solving algorithms. Both teams employed a similar pipeline, leveraging the systems developed for subtask 1 to generate candidate answers for each clue, then applying a search algorithm to fill the grid while enforcing crossing constraints. Hence, no team exploited the training and validation datasets released for subtask 2.

The FFT-UNIBA runs achieved by far the best results, reaching a character-level accuracy of 92% and correctly solving 34% of the grids in the test set. Given the higher performance obtained in Subtask 1, we hypothesize that the strength of the FFT-UNIBA clue-answering system, which acts as the main semantic bottleneck in the solving pipeline, plays a major role in the observed gap with UNIBA. Moreover, their multiple-seed strategy with high-confidence ranking, backtracking, and a constrained depth-first search

⁹An excerpt of the identified clusters, along with clue examples and the top terms extracted with TF-IDF are reported in Appendix A.

¹⁰Transl. “French city”, “male name”.

Systems	5×5			7×7			9×9			11×11			13×13		
	CA	WA	FM	CA	WA	FM	CA	WA	FM	CA	WA	FM	CA	WA	FM
FFT-UniBa_c1000_2	1.00	1.00	1.00	0.94	0.88	0.60	0.92	0.83	0.10	0.91	0.82	0.00	0.85	0.73	0.00
FFT-UniBa_c1000NODICT_2	1.00	0.98	0.90	0.95	0.90	0.60	0.92	0.85	0.10	0.91	0.81	0.00	0.84	0.73	0.00
FFT-UniBa_c1000_1	0.98	0.94	0.70	0.94	0.86	0.50	0.94	0.86	0.20	0.89	0.79	0.00	0.87	0.76	0.00
FFT-UniBa_c1000NODICT_1	0.97	0.90	0.70	0.91	0.82	0.30	0.93	0.84	0.10	0.91	0.81	0.00	0.85	0.74	0.00
UNIBA RUN2	0.89	0.77	0.40	0.85	0.72	0.30	0.83	0.68	0.10	0.80	0.65	0.00	0.70	0.52	0.00
UNIBA RUN1	0.86	0.72	0.40	0.86	0.72	0.30	0.85	0.69	0.10	0.80	0.65	0.00	0.71	0.52	0.00
Baseline	0.85	0.71	0.40	0.68	0.49	0.00	0.74	0.62	0.00	0.66	0.51	0.00	0.73	0.59	0.00

Table 3

Cruciverb-IT Subtask 2 results reported separately for each grid size (5×5, 7×7, 9×9, 11×11, 13×13).

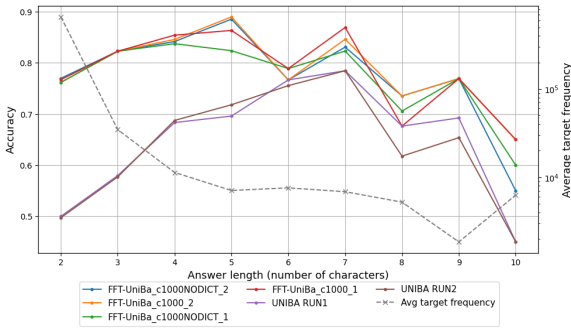


Figure 3: WA accuracies of each run across answer lengths (number of characters); the gray dashed-line marks the average target frequency.

System	Intersection number			
	1	2	3	4
FFT-UniBa_c1000_1	0.73	0.89	0.92	0.95
FFT-UniBa_c1000_2	0.74	0.88	0.92	0.95
FFT-UniBa_c1000NODICT_2	0.71	0.87	0.92	0.95
FFT-UniBa_c1000NODICT_1	0.72	0.87	0.91	0.95
UNIBA RUN2	0.60	0.70	0.81	0.88
UNIBA RUN1	0.60	0.72	0.81	0.89
Baseline	0.53	0.68	0.74	0.75

Table 4: Character accuracy by number of surrounding non-black cells (columns). Systems here are ranked by the overall CA.

approach proved effective in mitigating potentially incorrect early placements and, overall, appears well suited to the combinatorial nature of crossword solving, suggesting that explicit constraint propagation is crucial for grid-level reasoning.

A clear pattern emerges when analyzing performance across grid sizes in Table 3: while systems achieve near-perfect accuracy on 5×5 grids (up to 100% FM for the best system), performance degrades steeply as grid size increases, with the best model dropping from 100% to 60% FM when moving from 5×5 to 7×7 grids. No system achieved a complete solution on 11×11 or 13×13 grids, highlighting the exponential growth in complexity as the number of interdependent constraints increases.

Influencing Factors. For Subtask 2, the influence of lexical and structural factors on system performance appears less pronounced than in Subtask 1. In particular, the relationship between FullMatch accuracy and answer length is weaker. As shown in Figure 3, accuracy generally increases from answers of length 2 up to length 6, followed by a moderate decrease for longer answers, without the marked downward trend observed in Subtask 1. Overall, the drop in performance for longer words is noticeably less severe, and the curves across systems exhibit a smoother behavior. A similar pattern emerges when considering answer frequency. While average target frequency decreases across length bins, accuracy does not show a strong monotonic decline. Instead, performance initially increases and only decreases for the lowest-frequency bins, with a substantially milder effect compared to Subtask 1. A plausible explanation for these trends lies in the specific configuration of Subtask 2, which integrates a crossword grid solver into the prediction pipeline. In this setting, the final predictions do not solely reflect the behavior of the underlying neural models, but rather the interaction between the models and the solver. As a consequence, the solver may act as a filtering and re-ranking component, partially mitigating the impact of both lexical frequency and answer length, and thereby smoothing the correlations observed in Figure 2 (b). Moreover, the absence of a sharp performance drop for longer answers can be attributed to the presence of multiple grid constraints: although longer words are generally harder to predict in isolation, their instantiation within a crossword grid provides additional crossing letters and structural cues, which can facilitate their recovery with respect to Subtask 1.

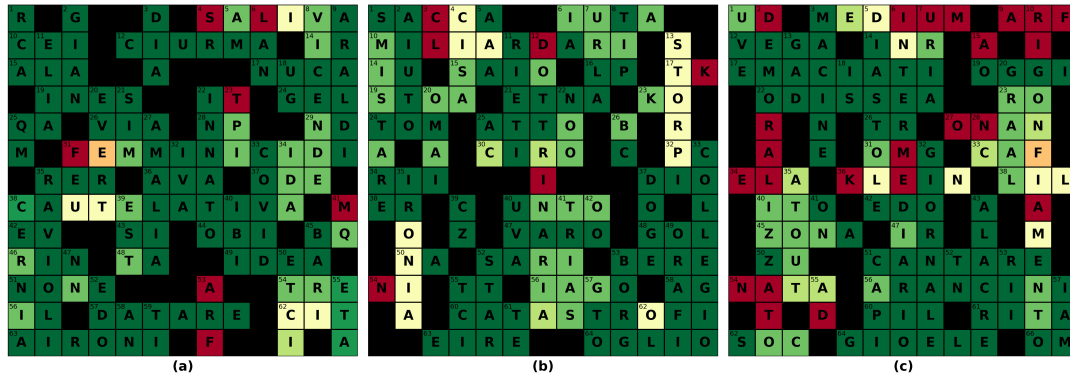


Figure 4: Example grids taken from the test set of subtask 2. The grids (see Appendix B for the corresponding clues) are populated with the correct answers and each cell is colored (green to red) by the fraction of systems that answered correctly (green cell = all systems predicted the correct character in that cell; red cell = no system predicted the correct character for that cell).

The Role of Intersections. Table 4 provides insight into why constraint-based approaches succeed: character accuracy increases monotonically with the number of intersecting words per cell (from 0.73 to 0.95 for FFT-UniBa). Cells with more intersections benefit from additional constraints that help disambiguate among candidate answers, effectively allowing the system to cross-check predictions. This finding aligns with human solving strategies, where solvers often rely on crossing words to confirm or reject candidate answers. Focusing in particular on the FFT-UniBa system, its multiple configurations reveal that larger candidate pools generally improve Full Match scores by increasing the likelihood of including the correct answer in the search space. Dictionary augmentation also provides modest but consistent improvements, particularly for rare words that may not appear in the model’s top predictions. We deem the trade-off between the gains from these approaches and the search and computational complexity an interesting topic for future research on efficient automatic crossword solvers.

Figure 4 shows three 13x13 crossword grids taken from the test set and populated by gold solutions with each cell colored by the fraction of systems that correctly predicted the corresponding character. These examples provide qualitative visual evidence supporting the trend reported in Table 4: systems tend to produce incorrect answers more frequently for isolated cells, particularly those corresponding to short entries or located near grid borders. For example, in Figure 4 crossword (c), no system was able to correctly fill numerous isolated cells such as the ones for clues 9, 15, 27, 34, 36, 54, 55 originating, often, regions of high density errors. Similar patterns are notable, also, in grids (a) and (b). This common trend highlights the performance advantage of constraint-based approaches when structural redundancy is available for disambiguation, while also underscoring the importance of a robust clue-answering expert for cases with limited or no intersection hints.

6.3. Human and Artificial Solvers

Looking at the core challenges encountered by the models in the Cruciverb-IT tasks, we have wondered whether such challenges are consistent with the difficulties perceived by human solvers. We have also asked ourselves whether the automatically created grids are somewhat in line with grids created by humans, and how they would be received by human creators and solvers. While running a full study with human subjects was not feasible at this time (although it might be considered in future extensions of this work), consultation with Italian crossword expert Stefano Bartezzaghi yielded some interesting considerations. We report them below.

Length At the level of the underlying models, longer solutions are generally harder to predict, with the exception of two-letter words, which often correspond to abbreviations, initials, or wordplay phenomena. This behaviour is consistent with the trends observed in Subtask 1, where predictions are produced in isolation. However, when considering the complete systems used in Subtask 2, i.e. the

combination of neural models and the grid solver, the role of answer length becomes less clear-cut. In this setting, the negative impact of longer words is attenuated, and performance tends to increase up to medium-length answers, followed by only a mild decrease for longer ones. This suggests that the constraints imposed by the crossword grid and the re-ranking performed by the solver partially compensate for the intrinsic difficulty of predicting longer strings in isolation. In particular, longer answers benefit from a higher number of crossings, which provide additional character-level constraints and can facilitate their recovery within the grid. Interestingly, this behaviour brings artificial systems closer to human solving strategies. For human solvers, longer words can in fact be easier to retrieve, due to the larger number of constraints and to the presence of standard bound morphemes, such as *-zione*, *post-*, *sub-*, *-abile*, *-mento*, which further restrict the set of plausible candidates and often make the solution more predictable. Moreover, longer and morphologically transparent words can help fill the grid more efficiently by constraining neighbouring entries. A similar distinction emerges for clue length. At the model level, longer clues tend to be more difficult to handle. From a human perspective, instead, difficulty is primarily determined by the degree of focus and ambiguity of the clue rather than by its length.¹¹ In fact, longer clues are often more informative and less ambiguous than short and highly compact ones, and can therefore be perceived as easier to solve.

Frequency At the level of the neural models, lexical frequency plays a major role: more frequent answers are generally more likely to be predicted correctly. This behaviour clearly emerges in Subtask 1, where systems operate without grid-level constraints. When considering the full systems employed in Subtask 2, however, the effect of frequency becomes weaker and less monotonic. Although frequent words still tend to be favoured, accuracy does not sharply decrease for low-frequency answers, and the overall trend appears substantially smoother. As in the case of answer length, this attenuation can be attributed to the presence of the grid solver, which integrates the predictions of the models with structural constraints derived from the crossword grid. As a result, the final output reflects the interaction between lexical preferences learned by the models and the combinatorial constraints enforced by the solver, rather than lexical frequency alone. This behaviour partially aligns artificial systems with human solving strategies. For human solvers, common words are in general easier to retrieve, but the specificity and ambiguity of the clue often play a more decisive role than raw lexical frequency. For instance, even a very frequent word such as *“albero”* (transl. “tree”) can become difficult to recover when the clue is vague (e.g. “vegetation”), semantically ambiguous (e.g. “Maestro di barca”), or relies on an idiosyncratic contextualization, that is, a clue grounded in a highly specific and unconventional frame rather than in a direct lexical relation (e.g. *“La sequoia lo è del mammut”*¹²). In this respect, the solver-based setting of Subtask 2 reduces the dominance of frequency-driven behaviour observed at the model level, and yields a performance profile that is closer to the human perception of difficulty.

Grid features While for models larger grids are very difficult, with no system being able to solve 11x11 and 13x13 crosswords in Cruciverb-IT, for human solvers the grid size per se does not play a direct role in the complexity of the task. One aspect instead which appears to be consistent across artificial and human solvers is cell isolation: the more neighbors a cell has, and therefore the more constraints, the easier it is to fill that with the correct character.

Do the grids feel “human”? As a general point, we have been wondering to what extent the generated grids resemble good examples of human-crafted crosswords and if they do come across as artificial. Apparently, the gap is there: in general, the grids we have generated would not be particularly

¹¹For instance, the solution *Torino* can be clued with increasingly specific definitions, such as *“città italiana”* (transl. “Italian city”), *“capoluogo di regione italiana”* (transl. “capital of an Italian region”), and *“capoluogo del Piemonte”* (transl. “capital of Piedmont”), where more detailed, and thus often longer definitions progressively reduce ambiguity and are typically perceived as easier by human solvers.

¹²Transl. “The sequoia is the mammoth’s one”. The clue relies on an implicit historical context: sequoias are extremely long-lived trees, and some specimens were already alive when mammoths still existed. The solution is therefore *“albero”*, which can only be retrieved by reconstructing this implicit and unconventional contextual relation.

appealing to a human solver, since they lack what crossword experts would call *rhythmic breadth*. This is due to the fact that the generated grids do not offer elaborate crossings and do not form large *squares*, except in very limited extensions.

7. Conclusion

We presented Cruciverb-IT, the first shared task on Italian crossword solving, held at EVALITA 2026. The task attracted five participating teams who submitted a total of 17 system runs across two subtasks: clue answering and full grid solving. We released a dataset of approximately 410,000 Italian clue-answer pairs and 600 automatically generated crosswords of varying sizes, providing a new benchmark for evaluating language understanding and reasoning capabilities in Italian.

Our evaluation reveals that modern NLP approaches achieve promising results in individual-clue answering, with the best system reaching 69% accuracy at rank 1 via retrieval-augmented LLM prompting. For grid solving, constraint-satisfaction methods combined with fine-tuned language models are most effective, achieving up to 92% character accuracy and solving 34% of grids completely. However, a clear scalability challenge emerges: while systems reliably solve smaller grids (5×5), the steep performance decrease in larger grids indicates that crossword solving at realistic scales remains an open problem.

In addition, the qualitative analysis reported in Section 6.3, based on a consultation with Italian crossword expert Stefano Bartezzaghi, highlights a systematic gap between human solving strategies and the behavior of current neural models, as well as a more nuanced picture when considering full crossword-solving systems. In particular, our observations show that factors such as morphological predictability, semantic focusing, ambiguity, and idiosyncratic contextualization of clues play a central role for human solvers. At the same time, model-level predictions are largely driven by surface-level properties such as answer length and lexical frequency, whereas system-level configurations that integrate a grid solver partially mitigate these effects by exploiting structural constraints induced by the crossword grid. This comparison suggests that future crossword-solving systems should explicitly account for higher-level linguistic and pragmatic properties of clues, and not only for grid-level constraints, in order to better approximate human strategies.

The findings suggest several directions for future research. First, hybrid architectures that combine the semantic understanding of Large Language Models with explicit constraint reasoning may be required to effectively improve grid-level performance, beyond solving individual clues. Second, the strong correlation between intersection density and accuracy suggests that iterative refinement strategies could be effective for progressively constraining the solution space. Finally, the resource and evaluation framework introduced in this task can serve as a blueprint for exploring crossword solving in other languages and for investigating the broader question of how language models handle linguistic puzzles at the intersection of cultural knowledge and logical reasoning.

Acknowledgments

We would like to thank Stefano Bartezzaghi for carefully analyzing our results and observations and for providing valuable insights that greatly helped us better interpret the linguistic and pragmatic aspects of the task. The authors acknowledge the support of the PNRR MUR project PE0000013-FAIR. Partial support was also received by the project “Understanding and Enhancing Preference Alignment in Large Language Models Through Controlled Text Generation” (IsCc8_ALIGNLLM), funded by CINECA under the ISCRA initiative, for the availability of HPC resources and support.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5 and Grammarly to conduct grammar and spelling checking. The author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

- [1] S. C. Shapiro, Artificial intelligence, in: S. C. Shapiro (Ed.), *Encyclopedia of Artificial Intelligence*, 2nd ed., John Wiley & Sons, Inc., New York, 1992, pp. 54–57.
- [2] M. L. Littman, Review: Computer language games, in: T. Marsland, I. Frank (Eds.), *Computers and Games*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 396–404.
- [3] M. Ernandes, G. Angelini, M. Gori, Webcrow: A web-based system for crossword solving, in: *AAAI Conference on Artificial Intelligence*, 2005. URL: https://link.springer.com/chapter/10.1007/11590323_37.
- [4] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, D. Klein, Automated crossword solving, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3073–3085. URL: <https://aclanthology.org/2022.acl-long.219>. doi:10.18653/v1/2022.acl-long.219.
- [5] J. Rozner, C. Potts, K. Mahowald, Decrypting cryptic crosswords: Semantically complex word-play puzzles as a target for nlp, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems*, volume 34, Curran Associates, Inc., 2021, pp. 11409–11421. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/5f1d3986fae10ed2994d14ecd89892d7-Paper.pdf.
- [6] S. Saha, S. Chakraborty, S. Saha, U. Garain, Language models are crossword solvers, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 2074–2090. URL: <https://aclanthology.org/2025.naacl-long.104/>.
- [7] A. Sadallah, D. Kotova, E. Kochmar, What makes cryptic crosswords challenging for LLMs?, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 5102–5114. URL: <https://aclanthology.org/2025.coling-main.342/>.
- [8] G. Angelini, M. Ernandes, M. Gori, Solving italian crosswords using the web, in: *International Conference of the Italian Association for Artificial Intelligence*, 2005. URL: https://link.springer.com/chapter/10.1007/11558590_40.
- [9] G. Barlacchi, M. Nicosia, A. Moschitti, A retrieval model for automatic resolution of crossword puzzles in italian language, in: *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & and of the Fourth International Workshop EVALITA 2014: 9-11 December 2014, Pisa, Pisa University Press*, 2014, pp. 33–37.
- [10] A. Moschitti, M. Nicosia, G. Barlacchi, SACRY: Syntax-based automatic crossword puzzle resolution sYstem, in: H.-H. Chen, K. Markert (Eds.), *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Beijing, China, 2015, pp. 79–84. URL: <https://aclanthology.org/P15-4014/>. doi:10.3115/v1/P15-4014.
- [11] G. Sarti, T. Caselli, M. Nissim, A. Bisazza, Non verbis, sed rebus: Large language models are weak solvers of Italian rebuses, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 888–897. URL: <https://aclanthology.org/2024.clicit-1.96/>.
- [12] G. Sarti, T. Caselli, A. Bisazza, M. Nissim, EurekaRebus - verbalized rebus solving with LLMs: A CALAMITA challenge, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 1202–1208. URL: <https://aclanthology.org/2024.clicit-1.132/>.
- [13] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, Crossword space: Latent manifold learning for italian crosswords and beyond, in: *Proceedings of the 11th Italian Conference on Computational*

- Linguistics (CLiC-it 2025), 2025.
- [14] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [15] M. L. Ginsberg, Dr. fill: Crosswords and an implemented solver for singly weighted csps, Journal of Artificial Intelligence Research 42 (2011) 851–886.
 - [16] K. Zeinalipour, T. Iaquina, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori, Italian crossword generator: Enhancing education through interactive word puzzles, in: Proceedings of the 9th Italian Conference on Computational Linguistics (CLiC-it 2023), 2023. URL: <https://ceur-ws.org/Vol-3596>.
 - [17] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, volume 4, Now Publishers Inc, 2009.
 - [18] S. Kulshreshtha, O. Kovaleva, N. Shivagunde, A. Rumshisky, Down and across: Introducing crossword-solving as a new NLP benchmark, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2648–2659. URL: <https://aclanthology.org/2022.acl-long.189/>. doi:10.18653/v1/2022.acl-long.189.
 - [19] L. De Moura, N. Bjørner, Z3: An efficient smt solver, in: International conference on Tools and Algorithms for the Construction and Analysis of Systems, Springer, 2008, pp. 337–340.
 - [20] A. Yassine, C. Savelli, D. Napolitano, G. Gallipoli, L. Cagliero, E. Baralis, Ac/dg at cruciverb-it: Retrieval-based approaches for italian crossword clue answering, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [21] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, et al., Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025).
 - [22] A. Porcelli, F. Di Gravina, E. Fontana, M. Curri, F. D. Di Gregorio, Fft-uniba at cruciverb-it: Special length tokens and csp for italian crossword solving, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [23] G. Sarti, M. Nissim, IT5: Text-to-text pretraining for Italian language understanding and generation, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 9422–9433. URL: <https://aclanthology.org/2024.lrec-main.823/>.
 - [24] F. Giobergia, Minds at cruciverb-it: Solving italian crossword clues with masked language models and candidate pooling, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [25] P. Basile, Uniba at cruciverb-it: Solving crosswords with encoder-decoder models and beam search, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [26] A. Shcherbakov, D. Croce, R. Basili, Unitor at cruciverb-it: Retrieval-augmented two-step reasoning for italian crossword clue answering, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [27] M.-L. M.-F. Multi-Granularity, M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.
 - [28] A. Zeng, X. Lv, Q. Zheng, Z. Hou, B. Chen, C. Xie, C. Wang, D. Yin, H. Zeng, J. Zhang, et al., Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, arXiv preprint arXiv:2508.06471 (2025).

A. Influencing Factors

Figure 5 shows the distribution of test instances across different system agreement levels, split according to whether the answer word appears in the training set. Table 5, instead, reports some examples of the clues and the corresponding top TF-IDF terms extracted by applying K-Means clustering to the set of instances incorrectly predicted by all the systems.

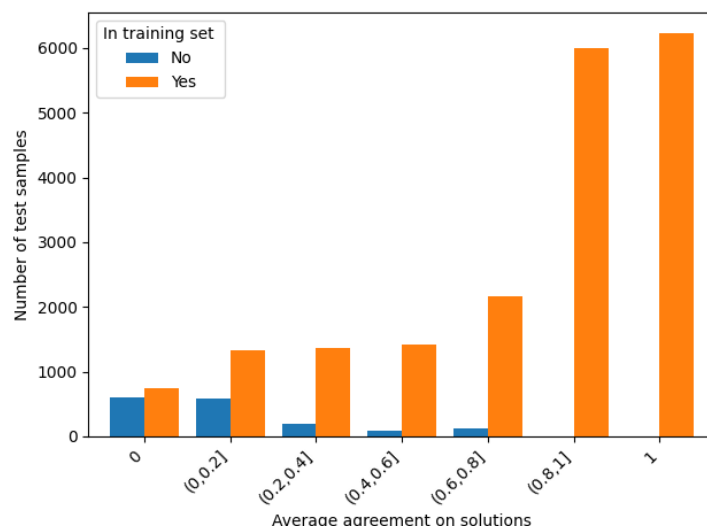


Figure 5: Distribution of test instances across system agreement levels, measured as the percentage of systems that correctly predicted the gold answer, split on whether the answer word appeared in the training set.

Cluster	Top terms	Clue examples
1	italiana, centro, tipo, re, successo, gioco, personaggio, noto, esserlo, piccola	La zona da cui nasce il Po; Malvagia; Si torna agli antichi; Tiranno... spagnolo; Condizione di chi ha due capi
2	famoso, scrittore, cantante, televisivo, storico, italiano, successo, soneria, soneria personaggio, tipo	Famoso dramma di Agatha Christie; Giornalista conduttore di un famoso Processo televisivo; Famoso scrittore; È famoso quello da Pro-cida; Famoso cantante italiano
3	nome, femminile, dà, famosa, tipo, televisivo, successo, scrittore, soneria, soneria personaggio	Nome d'uomo; Altro nome dei ricci di mare; Nome di molti spagnoli, messicani e argentini; Altro nome del martin pescatore; Danno nome a una famosa sonata di Beethoven
4	film, famoso, noto, personaggio, nome, televisivo, tipo, scrittore, soneria, soneria personaggio	Un famoso film a episodi; La musica che fa da sottofondo al film; È "onorario" quello d'un film con Richard Gere; Un film americano d'autore; Un noto film del regista Joseph Losey
5	celebre, televisivo, tipo, storico, soneria personaggio, soneria, successo, scrittore, roma, provincia	È diventato celebre quello di notte; La casa editrice di un celebre dizionarista; Un celebre Scipione; Una celebre massima latina relativa alla salute; Un celebre idillio di Manzoni
6	francese, famoso, tipo, storico, soneria personaggio, televisivo, successo, scrittore,	Città francese; Pierre, regista cinematografico francese; La Svizzera... francese; Regione francese; Guillaume-Léon, politico francese
7	donna, nome donna, nome, storico, tipo, televisivo, successo, scrittore, soneria, soneria	Una donna generosa; Donna ricaduta nel peccato, per la teologia; Attraente come una donna; Donna che non ci vede più; Donna di casa
8	provincia, comune provincia, comune, storico, tipo, televisivo, successo, scrittore, soneria, soneria personaggio	La località misteriosa di oggi (provincia di Bergamo); Comune in provincia di Salerno; Comune in provincia di Lecco; Comune in provincia di Verbania; Lombardi di provincia

Table 5

Examples of clusters obtained by applying TF-IDF + K-Means ($K = 10$) to the set of instances incorrectly predicted by all systems. Each row reports representative clue examples and the corresponding top TF-IDF terms characterizing the cluster.

B. Crossword Clues

We report here the crossword clues associated with the grids discussed in Section 6.2.

ACROSS

4. Il segreto di certe ghiandole
10. Una Pina del teatro
12. L'equipaggio di una nave pirata
14. Centro di mira
15. Tornante nel calcio
17. Fra capo e collo
19. Agnese a Toledo
22. Il centro di Positano
24. Permette di fissare e modellare i capelli
25. Quality Assurance
26. Quella di mezzo è preferibile
28. Le separa la O
29. Nel cuore delle Ande
31. Si susseguono spaventosamente
35. Metropolitana parigina
36. __, come lava! diceva Calimero
37. La si scrive per elogiare
38. La misura che si prende per tutelarsi
42. Le aveva in testa Eva
43. Lo dice chi accetta
44. Una cintura per arti marziali
45. Le consonanti dell'ubiquo
46. Precede... Tin-Tin
48. Coda di gusta
49. Quella fissa viene e non se ne va
51. Lo dicono i Romani... negando
54. Tante erano le Caravelle di Colombo
56. Articolo per muratore
57. Mettere giorno, mese e anno
62. Piccolo a Torino
63. Gli uccelli che possono essere cinerini

DOWN

1. Assicurazione per chi guida
2. Il primo nome di Volontè
3. Indaga sulla mafia
4. Iniziali dell'attore Rea
5. Il rimpianto Moro (iniz.)
6. Divisione amministrativa svedese
8. Si succedono nella vita
9. Leggevano in pubblico gli editti
11. Le sue gesta sono narrate nei due libri dei Re della
18. Ugo Gregoretti
20. Fu per anni la grande rivale della Navratilova
21. Comprano e vendono azioni
22. Insito, connaturato
23. Tribunale Penale Internazionale
25. Quantità Massima
27. La Rodriguez, regina del fado
31. Fuoco francese
32. Una voce in fattura
33. Il 19 responsabile della pandemia
34. L'attrice Di Benedetto
35. Hanno un gustoso ripieno
38. Grosso pesce da tana
39. La Cina gli è vicina
40. In quel luogo per Livio
41. Un'abbreviazione per... aree
45. Una ruota del Lotto
47. Le gemelle... di Ennio
50. Uno... starnuto
52. Il nome del ministro Ronchi
53. Il fumetto di un cane
55. Da tenera diventa avanzata
58. Il Nobel che ha dato il premio e non l'ha ricevuto (iniz.)
59. Segue erre ed esse

Table 6

Definitions and solutions for Grid (a).

ACROSS

1. Si porta sulle spalle
6. Si mette nel sacco
10. Sono pieni di quattrini
14. Il centro del Friuli
15. Abito di penitenza
16. Hanno preceduto i CD
17. Non si toccano in Fort Knox
19. L'...aula di Zenone
21. Alto vulcano siciliano
23. Lo può subire il pugile
24. Il nome dell'attore Selleck
25. Si può far di presenza
30. Un famoso re di Persia
32. Iniziali della Cruz
34. Canaletti veneziani
37. Ogni religione ha il suo
38. Simbolo dell'erbio
40. Cola sulla leccarda
47. Il battesimo della nave
48. Fa gridare i tifosi
50. Una patria della canzone italiana popolare (targa)
52. Lo indossa l'indiana
53. Un piacere da moderare
54. Venite in centro
55. La coppia in lotta
56. Recita un credo ateo
58. Grasso della critica (iniz.)
60. Eventi disastrosi

DOWN

1. Suddividere la posta
2. Si accorre per prestarlo
3. Il nucleo del nucleo
4. Noto passo appenninico
5. Precede le prime inserzioni dell'elenco
6. La rima della fattoria
7. Le lancia l'atterrito
8. Con "tap" forma un ballo
11. Provincia del Lazio
12. Così è l'adesione acritica a dogmi
13. C'è chi vorrebbe raddrizzar loro le gambe!
20. A questo punto... degli antichi vati
22. Prefisso per intelletto
25. Un po' acerbo
26. Le iniziali di Colleoni
33. La scuola con gli interni
35. Si apprezza nell'umorista
37. A lui spettava il governo della Repubblica di Venezia
39. Catanzaro
40. Il frutto che si mangia a chicco a chicco
41. Periodo geologico
42. Un risultato calcistico
49. Creano costosi monili
52. Un artista molto noto
53. Il famoso Björn del tennis
55. Pubblica guide (sigla)
57. Global Trade Organization
61. Si beve nel pomeriggio

Table 7

Definitions and solutions for Grid (b).

ACROSS

1. Le hanno il custode e la guardia
3. Sensitivo
9. Il fumetto di un cane
12. La più splendente stella della Lira
14. International Normalised Ratio (sigla)
17. Molto magri, denutriti
19. Era domani fino a ieri
22. Capolavoro ricordato con l'Iliade
23. La "erre" dei Greci
26. Nel centro di Montreal
27. Ha dato il nome a un "vizio" (che peraltro non aveva)
31. Orribili Mostriciattoli Genetici
33. Sportello di consulenza fiscale
34. Uno dei figli del biblico Caleb
36. Il pittore del blu
38. La Hardin pianista
40. Andato via
42. Il nome del ministro Ronchi
45. Porzione di territorio
47. Fanno di maggio un miraggio
50. Mezzo Zulù
51. Verbo... canoro
54. La neo... dal fiocco rosa
56. Manicaretta siciliani
60. Indice della ricchezza nazionale
61. L'attrice Tushingham
62. Società... in breve
64. Dix che non dipinge
66. Pari in forma

DOWN

1. Ci sono da tavola
2. Scoraggiato, avvilito
3. Ruote da mulino
5. Dilatazione delle cavità cardiache
6. Gli spetta una provvigione per quel che ha... combinato
7. Uccello oceanico delle zone artiche
10. Ingrossamento
13. Il Lerner dell'Infedele
15. Targa del Nord-Ovest
18. È bellissimo... in mezzo
20. Il sacro calice ricercato da Parsifal
28. In mezzo al cancan
32. Si compra in edicola
35. Una carta favorevole
41. Il titolo del deputato (abbreviazione)
43. Solerte e laborioso
51. Girano sui colli...
52. Iniziali di Nuvolari
53. Li percorrono le gondole
55. Preposizione articolata
57. Già latino, sono inglese

Table 8

Definitions and solutions for Grid (c).

AC/DG at Cruciverb-IT: Retrieval-Based Approaches for Italian Crossword Clue Answering

Ali Yassine^{1,†}, Claudio Savelli^{1,†}, Davide Napolitano^{1,†}, Giuseppe Gallipoli^{1,†}, Luca Cagliero¹ and Elena Baralis¹

¹Politecnico di Torino, Turin, Italy

Abstract

We present our submission to Task 1 of the Cruciverb-IT challenge at EVALITA 2026, which focuses on answering clues extracted from Italian crosswords. The task is framed as a constrained question-answering problem, to generate a ranked list of candidate solutions for each clue, given the expected answer length. To address this problem, we explore three complementary approaches: a lexical retrieval model based on BM25, a dense semantic retrieval model that relies on sentence-level embeddings, and a hybrid retrieve-and-rerank architecture that combines the outputs of the two retrieval strategies using a large language model as a judge. Experimental results on the official evaluation benchmark show that dense semantic retrieval achieves the highest overall performance, whereas both lexical and hybrid methods consistently outperform the official task baseline. These findings highlight the effectiveness of embedding-based representations for Italian crossword-clue answering and provide insights into the role of hybrid retrieval and reranking strategies in this task.

Keywords

Large Language Models, Information Retrieval, Crossword Solving, Italian Natural Language Processing

1. Introduction

“Opening remarks or preliminary section (12 letters)” – While it may be straightforward to guess that the answer is “introduction”, this is not always the case, particularly when dealing with clues that involve wordplay, ambiguity, lateral thinking, and cultural references, or, more generally, that require multiple reasoning steps and a higher level of abstraction in language understanding. For this reason, crossword puzzles have recently inspired the creation of new benchmarks in Natural Language Processing [1], as they require a unique synthesis of complex linguistic reasoning, open-domain knowledge retrieval, and strict constraint satisfaction. While earlier approaches mainly implemented rule- and retrieval-based solutions [2, 3], the widespread adoption of (Large) Language Models (LLMs), recently enriched with reasoning capabilities, has brought renewed attention to crossword solving and, more generally, to puzzle solving as challenging benchmarks [4].

The vast majority of research has been concentrated on the English language [5, 6], with few prior works addressing this task in non-English languages, such as Italian [7]. For this reason, we conduct our research to propose and investigate solutions for Italian crossword solving in the context of the Cruciverb-IT task [8] at EVALITA 2026 [9]. The proposed shared task comprises two subtasks: *Task 1*, which consists of answering clues given the clue text and the length of the target answer, without any additional grid constraints; and *Task 2*, which extends the previous task by requiring the solution of complete Italian crossword grids.

Despite their success across many scenarios, Large Language Models still suffer from known limitations in handling wordplay, culture-specific knowledge, or sub-token counting issues [10, 11, 5]. Furthermore, the challenging nature of crossword solving is exacerbated in Romance languages such

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

[†]Corresponding authors. These authors contributed equally.

✉ ali.protect@TU_yassine@polito.it (A. Yassine); claudio.savelli@polito.it (C. Savelli); davide.napolitano@polito.it (D. Napolitano); giuseppe.gallipoli@polito.it (G. Gallipoli); luca.cagliero@polito.it (L. Cagliero); elena.baralis@polito.it (E. Baralis)

ORCID 0009-0004-3517-8754 (A. Yassine); 0009-0005-1108-1170 (C. Savelli); 0000-0001-9077-4103 (D. Napolitano); 0009-0003-1744-6674 (G. Gallipoli); 0000-0002-7185-5247 (L. Cagliero); 0000-0001-9231-467X (E. Baralis)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

as Italian. In these languages, rich morphology, gender–number agreement, and high inflection rates introduce greater entropy into the answer-generation process. While the research community has moved toward tackling “*extreme ambiguity*” in English cryptic crosswords [11, 12], Italian solvers must face a different set of complexities that differ significantly from the English-centric training data dominating current LLM pre-training. As an example, consider the English clue “*Fast (5 letters)*”, which would map relatively straightforwardly to “*quick*” or “*rapid*”. In contrast, the corresponding Italian clue, for example “*Veloce (6 lettere)*”, would lead to a much larger set of possible solutions, such as “*celere*”, “*fugace*”, “*presto*”, “*rapido*”, “*pronto*”, and “*svelto*”, as well as inflectional variants like “*rapida*” and “*pronta*” to account for feminine gender agreement. This highlights that, compared to English, Italian exhibits both greater lexical variety and richer morphology due to gender-number agreement and other morphological processes (e.g., verb conjugation and derivational modification). On the other hand, richer morphology may impose tighter constraints during the grid-filling phase, thereby narrowing the solution space. Considering the previous example, if a crossing entry determines that the final letter is ‘*E*’, the set of potential solutions listed above can be reduced to only the first two candidates.

This paper presents the solutions developed by the AC/DG team¹ to the Cruciverb-IT Task 1, including an LLM-based approach that exploits reasoning capabilities, a more traditional retrieval-based solution, and a combination of the two. The first method leverages lightweight LLM fine-tuning on synthetic reasoning traces to train the model to reason over the provided clue and generate the final answer. In the second approach, a retrieval-based solution is implemented that investigates both lexical- and semantic-level similarity as retrieval strategies. Lastly, we employ an LLM as a generative reranker that, given the outputs of the retrieval-based method, produces a final ordered list of candidate answers. The results confirm the difficulties LLMs face in addressing this task [6, 12], whereas simpler retrieval-based solutions achieve better performance.

2. Background and Related Work

Large Language Models have recently been employed and evaluated across a wide range of tasks that require reasoning. When referring to “*reasoning*”, although there is no unique definition of the term [13], the most common benchmarks entail arithmetic, commonsense, or symbolic reasoning. However, in practice, many other reasoning tasks exist, such as causal judgment, temporal understanding, and word sorting [13]. Interestingly, the research community has recently highlighted puzzle solving as a promising testbed for evaluating the applicability of LLMs to complex reasoning tasks [4]. Among rule-based puzzles, deterministic games include crossword solving.

Unlike more narrowly focused tasks such as arithmetic reasoning that do not typically involve broad and diverse semantic knowledge, crossword puzzle solving demands multifaceted skills and knowledge. This is because crossword solving can be viewed as a form of open-domain question answering, requiring the interpretation and resolution of a wide range of clues that draw on substantial linguistic and real-world knowledge. Additionally, the generated answers must also satisfy the constraints of the crossword grid. According to Kulshreshtha et al. [1], crossword clues can be categorized into multiple classes, including factual, historical, abbreviations, word meanings, synonyms and antonyms, and wordplay.

The field of automated crossword solving has advanced significantly, addressing the distinct challenges posed by American-style (i.e., straight) crosswords and British-style (i.e., cryptic) crosswords. For American-style puzzles, which rely heavily on grid constraints and knowledge retrieval (e.g., “*Meryl of movies (5 letters)*”, with the answer “*Streep*”), Wallace et al. [10] propose a system that combines neural question answering with belief propagation and local search. Building on this, Saha et al. [5] recently demonstrated that general-purpose LLMs can achieve high performance by leveraging their reasoning capabilities alongside a constraint-based search. Regarding cryptic crosswords, which demand complex linguistic disambiguation and wordplay interpretation (e.g., “*Bird colour (4 letters)*”, with the answer “*teal*”, which is both a colour and a type of bird), Efrat et al. [12] found that standard fine-tuned models such as T5 struggle to surpass rule-based solvers. Rozner et al. [11] proposed a curriculum learning

¹From the first letters of the names of the first four authors.

approach, while Sadallah et al. [6] evaluated the zero- and few-shot capabilities of modern LLMs on these tasks, observing that models still lag behind human experts due to difficulties in decomposing wordplay and understanding complex clues.

Due to the abundance of resources in English, puzzle-solving has only recently attracted interest in other languages, such as Italian. For example, Sarti et al. [14] evaluate LLM performance on rebus resolution. With respect to crossword puzzles, the task of crossword generation has been explored for educational purposes [15, 16] by leveraging LLMs to automatically generate pedagogical clues from Wikipedia pages. Italian crossword solving has been addressed both through traditional retrieval-based approaches and by exploiting language model capabilities. Specifically, Angelini et al. [2] propose a framework integrating database, rule-based, dictionary, and web search modules to generate candidate solutions, which are then filtered and merged. In [3], a database is constructed and queried, combining search engine scores with statistical similarity features. Lastly, Ciaccio et al. [7] model the problem as a retrieval task while leveraging siamese and asymmetric dual encoder architectures to capture the underlying relations between crossword clues and their solutions.

In our work, inspired by recent advancements in LLM reasoning for crossword solving and by the lack of evaluation for this task in Italian, we first propose an LLM-based approach. However, prior work has observed that LLMs struggle with linguistic and culturally-relevant knowledge [7]. This limitation can be particularly critical for cryptic crosswords, especially for clues that require ambiguity or polysemy resolution and deeper semantic understanding, such as those involving wordplay, anagrams, nonliteral language, and culture-specific knowledge. These challenges are confirmed by known limitations of LLMs in handling figurative language [17] or non-standard lexical and morphological variation, potentially involving language varieties [18]. For this reason, we also investigate a retrieval-based approach, which may better handle knowledge-grounded clues. Finally, to exploit the strengths of both strategies, we also propose a hybrid approach that combines the two solutions.

3. Data and Task Setup

Challenge Overview. Cruciverb-IT at EVALITA 2026 consists of two tasks. Task 1 focuses on *Italian crossword clue answering*, while Task 2 involves *autonomously solving Italian crossword grids*. In this work, we participate in Task 1.

Task Definition. The first task of the challenge focuses on *Italian crossword clue answering*. The task is formulated as a question-answering problem, where a system must predict one or more candidate solutions for a given crossword clue. Formally, let’s consider a set of clues $C = \{c_1, c_2, \dots, c_n\}$. For each clue c_i , the method must produce a ranked list of candidate answers $S = \{s_1, s_2, \dots, s_k\}$, sorted by decreasing likelihood, where the correct solution s_i may appear at any position in the list. To better approximate a realistic crossword-solving scenario and to constrain the search space, each clue is paired with the character length of the target answer. Systems are therefore required to generate candidate answers that respect the specified length constraint. The use of external data sources that explicitly contain crossword clues is prohibited to avoid simple lookup-based systems.

An example input consists of a clue and a target length, such as: “*Lo puoi trovare a tavola (4 lettere)*” (English: “You can find it on the table”). Given this input, a system may generate a ranked list of candidate answers such as $\{“pane” (bread), “vino” (wine), “sale” (salt), “olio” (oil), \dots\}$, ideally including the correct answer “*pane*”.

The methods are evaluated using ranking-based metrics. Specifically, Accuracy@1 and Accuracy@10 measure the proportion of clues for which the correct answer appears among the top 1 or top 10 ranked candidate solutions, respectively. Additionally, Mean Reciprocal Rank (MRR) is computed as the average of the reciprocal ranks of the first correct answer across all evaluated clues.

Baseline System. The official baseline treats crossword-clue answering as an information retrieval problem. Given a test clue $c_i \in C_{\text{test}}$, the system computes a similarity score between c_i and each clue in the training set C_{train} . The baseline selects the 10 most similar training clues and returns their associated answers as candidate solutions, ranked by Best Matching 25 (BM25) similarity scores.

Dataset. The dataset used for the task combines the ItaCW crossword dataset [19] with an additional collection of clue–solution pairs gathered from web sources. The dataset contains approximately 410,000 unique clue–answer pairs and covers a wide range of crossword clue types, including wordplay-based, cryptic, named-entity initials, and fill-in-the-blank clues.

4. Methodology

The methodology for Task 1 of the Cruciverb-IT challenge is structured around the comparison of three distinct architectural paradigms for solving crossword clues. Rather than treating each solution in isolation, our framework moves from independent retrieval methods to a unified hybrid system. All systems strictly adhere to the competition rules by utilizing only the provided training data C_{train} and general-purpose linguistic resources, without accessing external crossword-specific databases.

4.1. Reasoning-Based LLM Fine-Tuning (Exploratory)

As an initial approach, we investigated whether Italian crossword solving could be addressed as an end-to-end generative task by explicitly teaching a model how to reason over clues. To this end, we used the Qwen3 8B model² to generate structured reasoning traces by conditioning on pairs of crossword clues and their gold answers. The model was prompted to produce step-by-step solution rationales that capture linguistic, semantic, or cultural reasoning patterns leading to the answer, while ensuring that the response adhered to the exact answer length and the principles of progressive discovery.

These generated rationales were then used to fine-tune the same Qwen3 8B model on this synthetic reasoning corpus, to internalize recurring crossword-solving strategies and generalize them to unseen clues.

Preliminary experiments indicated that this approach was poorly aligned with the task’s constraints. The model struggled to consistently reproduce the exact lexical forms required by the crossword grid and to recall culturally specific knowledge in a fully generative setting reliably. For this reason, we did not pursue this paradigm further and focused on retrieval-based methods as the core solution. The full Qwen3 8B prompt used to generate reasoning traces is provided in Appendix A.

4.2. Retrieval-Based Approaches

Based on the limitations observed in fully generative reasoning, we adopted retrieval-based methods to exploit observed clue–answer pairs. Our retrieval framework encompasses three components: sparse lexical retrieval, dense semantic retrieval, and a hybrid ensemble that leverages an LLM to rerank candidate answers, as described below.

4.2.1. Solution 1: Lexical Retrieval via BM25

The first solution leverages a sparse retrieval framework based on the BM25 ranking function. This approach is optimized to capture explicit lexical overlaps, serving as an effective mechanism for definitional clues. To improve retrieval efficiency and precision, we implement a **Length-Constrained Indexing** strategy.

- **Bucketed Search Space:** Let $\mathcal{D}$ be the unified training and validation corpus. We partition the corpus into disjoint buckets by answer length L . For a query q , the search is restricted to the specific bucket $\mathcal{B}_L = \{d \in \mathcal{D} : \text{len}(s_d) = L\}$. If $\mathcal{B}_L = \emptyset$, the system defaults to a global search across $\mathcal{D}$.
- **Exact Scoring Logic:** For each document $d \in \mathcal{B}_L$, the system calculates a relevance score by iterating over the query tokens $t \in q$. The score is computed using the inverse document

²<https://huggingface.co/Qwen/Qwen3-8B>

frequency (IDF) and a normalized term frequency component:

$$\text{score}(d, q) = \sum_{t \in q} \text{IDF}(t) \cdot \frac{f(t, d) \cdot (k_1 + 1)}{f(t, d) + k_1 \cdot (1 - b + b \cdot \frac{|d|}{\text{avgdl}})} \quad (1)$$

where $f(t, d)$ represents the raw count of term t in document d , avgdl is the average length of documents in the collection, and $|d|$ is the token count of the clue. k_1 and b are two hyperparameters that are set to their default values of 1.5 and 0.75, respectively. This implementation ensures that unique and highly descriptive keywords in a clue carry more weight in the final ranking.

4.2.2. Solution 2: Dense Retrieval in Latent Semantic Space

The second approach builds on the observation that handling morphological variation and out-of-vocabulary terms may be relevant in Italian crossword solving, where inflected and less frequent word forms are common [20]. This is done by mapping clues into a continuous latent space $\mathbb{R}^n$. This handles conceptual associations that lack token-level overlap.

- **Transformer-based Encoding:** We employ a Sentence-BERT (SBERT) architecture fine-tuned on Italian corpora³. The model functions as an encoder $E : \text{text} \rightarrow \mathbb{R}^{768}$, producing a dense vector representation $\mathbf{v} = E(\text{clue})$.
- **Geometric Search and Length Pruning:** Consistent with the lexical approach, the retrieval process is strictly governed by the target length L . For a test query q , we generate an embedding $\mathbf{q}$. The search is restricted to the subset of pre-computed embeddings $\{\mathbf{v}_j\}$ where the corresponding answer s_j satisfies $\text{len}(s_j) = L$.
- **Metric Formulation:** Proximity is quantified via the cosine similarity, measuring the distance between the query and candidate vectors:

$$S_{\cos}(\mathbf{q}, \mathbf{v}_j) = \frac{\mathbf{q} \cdot \mathbf{v}_j}{\|\mathbf{q}\| \|\mathbf{v}_j\|} \quad (2)$$

4.2.3. Solution 3: Generative Reranking via Heuristic Fusion

The third solution implements a hybrid **Retrieve-and-Rerank** strategy. This approach uses Qwen3 8B as a judge to integrate lexical and semantic evidence by synthesizing information from candidate sets generated by BM25 and dense retrieval models. The judge operates in a zero-shot configuration, without task-specific fine-tuning.

1. **Candidate Selection:** For each clue, the top candidate from BM25 lexical retrieval and the top candidate from dense retrieval are selected and presented to the judge. Together, these define a high-recall candidate pool $\mathcal{P}$ for evaluation.
2. **Evaluation and Reranking:** The judge evaluates all candidates and produces a reranked list according to plausibility, placing the most likely solution at the top.
3. **Generative Fallback:** If none of the candidates is deemed satisfactory, the judge may generate a new solution, which is incorporated at the top of the reranked list.
4. **Final Output:** The highest-ranked answer is selected according to the judge’s evaluation, with candidates presented in descending order of plausibility.

The full prompt used to guide the model is provided in Appendix B.

³<https://huggingface.co/nickprock/sentence-bert-base-italian-xxl-uncased>

5. Results and Discussion

The evaluation of the proposed systems was conducted using the official metrics for Task 1: Accuracy at 1 (Acc@1), Accuracy at 10 (Acc@10), and Mean Reciprocal Rank (MRR). Table 1 provides a comprehensive comparison of our three implementations against the official Cruciverb-IT baseline, including the relative performance gain (Δ) achieved by our optimal configuration.

Table 1

Comparative performance and relative gain (Δ) of our solutions over the official task baseline.

System Configuration	Acc@1 (Δ)	Acc@10 (Δ)	MRR (Δ)
Official Task Baseline	0.394 (—)	0.612 (—)	0.457 (—)
Solution 1: Optimized BM25	0.470 (+19.3%)	0.670 (+9.5%)	0.530 (+16.0%)
Solution 2: Dense Embeddings	0.510 (+29.4%)	0.730 (+19.3%)	0.570 (+24.7%)
Solution 3: Hybrid Qwen3 Judge	0.460 (+16.8%)	0.690 (+12.8%)	0.520 (+13.8%)

The results indicate that all three solutions improve upon the official Cruciverb-IT Task 1 baseline. Solution 2, based on dense semantic retrieval, achieves the highest performance across all metrics, with a relative gain of 29.4% in Acc@1. This demonstrates the effectiveness of embedding-based similarity for capturing semantic and conceptual relationships in the dataset. Solution 1, which relies on optimized BM25 lexical retrieval, also shows substantial improvements, particularly when test-set clues overlap lexically or are closely related to those in the training set. This suggests that retrieval-based approaches are particularly useful for datasets with repeated or highly similar clues, thereby exploiting redundancy in the available training data.

Solution 3 introduces a hybrid reranking approach in which a zero-shot Qwen3 8B model serves as a judge for the candidates from Solutions 1 and 2. Its overall accuracy is lower than dense retrieval, which can be explained by a few factors: the judge was not fine-tuned for the task, it has to choose between candidates that may compete with each other, and the dataset includes many ambiguous or nearly identical clues, which can cause the judge to favor plausible but incorrect “distractor” terms. Despite this, the judge still offers conceptual value by reranking candidates based on plausibility and providing a fallback generation when both top candidates are incorrect. This shows that generative models can combine evidence from different sources, even without task-specific training.

Overall, the results highlight the strong effectiveness of retrieval-based approaches on this dataset, especially when lexical or semantic overlap exists between training and test clues. While generative evaluation did not improve accuracy here, it indicates a promising research direction for integrating retrieval- and reasoning-based models in structured, knowledge-grounded tasks.

6. Conclusion and Future Work

In this paper, we present our submission to Task 1 of the Cruciverb-IT challenge at EVALITA 2026, treating Italian crossword-clue answering as a constrained question-answering problem. We evaluated three complementary approaches: a lexical retrieval model based on BM25, a dense semantic retrieval model using sentence-level embeddings, and a hybrid retrieve-and-rerank architecture that combines lexical and dense candidates via a Large Language Model serving as a judge.

Experimental results show that all proposed approaches outperform the official baseline, confirming the effectiveness of retrieval-based formulations for Italian crossword clue answering. Among the evaluated methods, dense semantic retrieval achieves the strongest overall performance across all ranking-based metrics, indicating that embedding-based representations are more effective at capturing the semantic associations required by Italian crossword clues.

Future work may explore a more fine-grained analysis of system behavior across different categories of clues, such as definitional, cryptic, or wordplay-based clues, to better understand the limitations of current retrieval-based methods. Furthermore, since some types of clues require identifying and

following predefined patterns to derive the answer (e.g., clues that implicitly refer to the letters at the beginning, middle, or end of a given word), another research direction could involve employing alternative loss functions beyond the traditional language modeling objective to fine-tune models, with the goal of better constraining their generation and enforcing closer adherence to the structure of the clue and the expected answer [21]. Finally, extending the proposed approaches to the complete crossword grid completion task represents a natural direction for further work, requiring joint reasoning over multiple clues and grid-level constraints.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT 5.2 for proofreading and formatting assistance. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] S. Kulshreshtha, O. Kovaleva, N. Shivagunde, A. Rumshisky, Down and across: Introducing crossword-solving as a new NLP benchmark, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2648–2659. URL: <https://aclanthology.org/2022.acl-long.189/>. doi:10.18653/v1/2022.acl-long.189.
- [2] G. Angelini, M. Ernandes, M. Gori, Solving italian crosswords using the web, in: S. Bandini, S. Manzoni (Eds.), *AI*IA 2005: Advances in Artificial Intelligence*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 393–405.
- [3] G. Barlacchi, M. Nicosia, A. Moschitti, A retrieval model for automatic resolution of crossword puzzles in italian language, in: *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & the Fourth International Workshop EVALITA 2014: 9-11 December 2014, Pisa, Pisa University Press, 2014*, pp. 33–37.
- [4] P. Giadikiaroglou, M. Lymperaio, G. Filandrianos, G. Stamou, Puzzle solving using reasoning of large language models: A survey, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 11574–11591. URL: <https://aclanthology.org/2024.emnlp-main.646/>. doi:10.18653/v1/2024.emnlp-main.646.
- [5] S. Saha, S. Chakraborty, S. Saha, U. Garain, Language models are crossword solvers, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 2074–2090. URL: <https://aclanthology.org/2025.naacl-long.104/>. doi:10.18653/v1/2025.naacl-long.104.
- [6] A. Sadallah, D. Kotova, E. Kochmar, What makes cryptic crosswords challenging for LLMs?, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 5102–5114. URL: <https://aclanthology.org/2025.coling-main.342/>.
- [7] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, Crossword space: Latent manifold learning for Italian crosswords and beyond, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings*, Cagliari, Italy, 2025, pp. 245–255. URL: <https://aclanthology.org/2025.clicit-1.26/>.
- [8] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, M. Nissim, Cruciverb-it @ evalita 2026: Overview of the crossword solving in italian task, in: *Proceedings of the Ninth Evaluation Campaign of Natural*

Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [9] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [10] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, D. Klein, Automated crossword solving, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3073–3085. URL: <https://aclanthology.org/2022.acl-long.219/>. doi:10.18653/v1/2022.acl-long.219.
- [11] J. Rozner, C. Potts, K. Mahowald, Decrypting cryptic crosswords: Semantically complex word-play puzzles as a target for nlp, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 11409–11421. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/5f1d3986fae10ed2994d14ecd89892d7-Paper.pdf.
- [12] A. Efrat, U. Shaham, D. Kilman, O. Levy, Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 4186–4192. URL: <https://aclanthology.org/2021.emnlp-main.344/>. doi:10.18653/v1/2021.emnlp-main.344.
- [13] J. Huang, K. C.-C. Chang, Towards reasoning in large language models: A survey, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1049–1065. URL: <https://aclanthology.org/2023.findings-acl.67/>. doi:10.18653/v1/2023.findings-acl.67.
- [14] G. Sarti, T. Caselli, M. Nissim, A. Bisazza, Non verbis, sed rebus: Large language models are weak solvers of Italian rebuses, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 888–897. URL: <https://aclanthology.org/2024.clicit-1.96/>.
- [15] K. Zeinalipour, T. Iaquina, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori, Italian crossword generator: Enhancing education through interactive word puzzles, in: F. Boschetti, G. E. Lebani, B. Magnini, N. Novielli (Eds.), Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2023), CEUR Workshop Proceedings, Venice, Italy, 2023, pp. 455–464. URL: <https://aclanthology.org/2023.clicit-1.55/>.
- [16] A. Zugarini, K. Zeinalipour, S. S. Kadali, M. Maggini, M. Gori, L. Rigutini, Clue-instruct: Text-based clue generation for educational crossword puzzles, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), ELRA and ICCL, Torino, Italia, 2024, pp. 3347–3356. URL: <https://aclanthology.org/2024.lrec-main.297/>.
- [17] G. Gallipoli, L. Cagliero, It is not a piece of cake for GPT: Explaining textual entailment recognition in the presence of figurative language, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 9656–9674. URL: <https://aclanthology.org/2025.coling-main.646/>.
- [18] G. Gallipoli, M. L. Quatra, D. R. Cambrin, S. Greco, L. Cagliero, DANTE at geolingit: Dialect-aware multi-granularity pre-training for locating tweets within italy, in: Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), Parma, Italy, September 7th-8th, 2023, volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: <https://ceur-ws.org/Vol-3473/paper14.pdf>.
- [19] K. Zeinalipour, T. Iaquina, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori, Italian crossword generator: Enhancing education through interactive word puzzles (2023).

- [20] C. Savelli, F. Giobergia, Enhancing cross-lingual word embeddings: Aligned subword vectors for out-of-vocabulary terms in fasttext, in: 2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT), IEEE, 2024, pp. 1–6.
- [21] D. Rege Cambrin, G. Gallipoli, I. Benedetto, L. Cagliero, P. Garza, Beyond accuracy optimization: Computer vision losses for large language model fine-tuning, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 12060–12079. URL: <https://aclanthology.org/2024.findings-emnlp.704/>. doi:10.18653/v1/2024.findings-emnlp.704.

A. LLM Prompt for Reasoning Trace Generation

The following prompt was used to generate structured step-by-step reasoning traces from crossword clues and their corresponding answers with Qwen3 8B model. These traces formed the synthetic dataset for our exploratory reasoning-based approach.

Sei un assistente AI che genera dati per un dataset di addestramento.
Il tuo compito è generare il ragionamento passo-passo che porta dall'indizio alla risposta data.
Il ragionamento deve essere procedurale, senza salti da supposizioni a parole inerenti alla risposta.
Il contesto è quello dei cruciverba in lingua italiana.

Indizio: "{clue}"

Risposta: "{answer}"

REGOLE FONDAMENTALI:

1. NON menzionare la "Risposta Target" ("{answer}") prima dell'ultimo passaggio.
2. Il ragionamento deve sembrare una scoperta progressiva.
3. La risposta finale deve essere una parola di esattamente {answer_length} lettere.

ISTRUZIONI PER IL RAGIONAMENTO:

1. Inizia analizzando l'indizio: "{clue}".
2. Se scomponi le parole dell'indizio, soffermati solo sugli elementi principali (non considerare articoli, ecc).
3. Analizza come è formulato l'indizio (es. metafora, abbreviazione, doppio senso, ecc).
4. Spiega il ragionamento, il gioco di parole, o la connessione logica che porta dall'indizio alla risposta.
5. Prova a fare ipotesi multiple, scartando quelle che non calzano.
6. Tieni presente che la risposta finale deve essere una parola di esattamente {answer_length} lettere.
7. Solo alla fine, come conclusione del tuo pensiero, identifica la risposta corretta.

STRUTTURA OBBLIGATORIA DEL RAGIONAMENTO:

Devi formattare il tuo output seguendo esattamente questi 4 punti:

1. ****Analisi dell'Indizio****: Analizza le parole chiave dell'indizio letteralmente e figurativamente.
2. ****Generazione Ipotesi****: Proponi possibili significati, giochi di parole o contesti (es. geografia, oggetti, modi di dire).
3. ****Verifica e Filtro****: Scarta le ipotesi che non calzano perfettamente e focalizzati su quella più logica.
4. ****Conclusione****: Dichiarare la risposta finale in modo definitivo.

B. LLM Prompt for Generative Reranking

Below is the Italian-language prompt used for the Qwen3 8B model in Solution 3. This prompt defines the evaluation procedure and the expected output format for the candidate solutions.

Sei un valutatore imparziale che confronta due risposte di modelli, {top_ans_1} (dal modello 1) e {top_ans_2} (dal modello 2), e il loro insieme combinato di risposte candidate: {candidates_all}.

Valuta tutte le risposte rispetto alla seguente domanda.

Domanda: {clue}

Lunghezza della risposta prevista: {answer_length} lettere

Procedura di valutazione

1. Valuta individualmente le risposte {top_ans_1} e {top_ans_2}.

- Determina se una delle due risposte risponde correttamente alla domanda.
- Se esattamente una tra {top_ans_1} o {top_ans_2} è corretta, restituisci quella risposta come risposta_finale.
- Se entrambe le risposte sono ugualmente corrette, restituisci quella risposta come risposta_finale.
- Se entrambe le risposte sono errate, verifica se una delle risposte candidate è corretta.
- Se almeno una risposta candidata è corretta, scegli la migliore come risposta_finale.

2. Soltanto se nessuna delle risposte candidate è corretta, genera una nuova risposta corretta:

- La nuova risposta deve rispondere direttamente alla domanda e rispettare la lunghezza prevista ({answer_length} lettere).
- Restituisci questa nuova risposta come risposta_finale.

3. Ordinamento delle risposte candidate

- Ordina tutte le risposte candidate dalla più probabile alla meno probabile di essere corretta.

Formato di output:

risposta_finale: <la risposta selezionata o appena generata>

risposte_candidate:

1. <risposta candidata più probabile>

2. <seconda migliore candidata>

...

N. <risposta candidata meno probabile>

FFT-UniBa at Cruciverb-IT: Special Length Tokens and CSP for Italian Crossword Solving

Andrea Porcelli^{1,†}, Filippo Di Gravina^{1,†}, Emanuele Fontana¹, Mattia Curri^{1,*} and Francesco Damiano Di Gregorio¹

¹Department of Computer Science, University of Bari Aldo Moro, Via Edoardo Orabona 4, 70125, Bari, Italy

Abstract

Cruciverb-IT [1], a shared task proposed at EVALITA 2026 [2], aims to encourage research in automated crossword solving systems by providing a challenging testbed for developing and evaluating systems focused on crossword puzzle solving. The proposed sub-tasks are the following: answering clues extracted from Italian crosswords and autonomously solving Italian crossword grids. To succeed in crossword solving, models must acquire meta-linguistic abilities beyond pure semantic understanding, like counting individual characters or enforcing strict length and pattern constraints typical of crossword answers. In this paper, we present our approach to both sub-tasks, which involves fine-tuning an Italian T5 model with special tokens to represent answer lengths and employing a constraint satisfaction problem (CSP) solver for grid filling. The system demonstrates competitive performance on both tasks, showcasing the effectiveness of our methods in mitigating the challenges posed by crossword puzzles. The code, and any additional resources used in our contribution can be found at <https://github.com/mattiacurri/fftuniba-cruciverbit>.

Keywords

Language Games, Crossword Solving, Constraint Satisfaction Problem, Natural Language Processing, Information Retrieval, Computational Linguistics

1. Introduction

Crosswords Puzzles are the most famous language games played around the world [3]. Among popular language games, crossword puzzles stand out as particularly challenging, demanding not only linguistic competence but also extensive world knowledge, cultural awareness, and lateral thinking skills [4]. Despite the impressive advancements in Large Language Models (LLMs), their performance on language games such as crosswords remains limited, especially in morphologically rich and less-resourced languages like Italian [5, 6, 7], where crosswords are characterized by deeper and more ambiguous references to socio-cultural and political contexts [8]. To succeed in solving crosswords, models must acquire meta-linguistic abilities beyond pure semantic understanding, such as counting individual characters or enforcing strict length and pattern constraints typical of crossword answers [9, 10, 11]. Because common tokenization schemes operate at word, subword, or byte-level granularity, character-level properties are not always preserved or directly accessible, so models cannot rely on token counts to satisfy orthographic constraints [12]. To be robust, systems should either learn to incorporate explicit character-aware representations, reconstruct character sequences from tokens, or be trained with character-level supervision or auxiliary objectives that expose and enforce these constraints [13]. To evaluate these capabilities in a standardized setting, the Cruciverb-IT shared task [1] was proposed within the EVALITA 2026 campaign [2]. The challenge establishes a benchmark for Italian crossword solving, dividing the problem into two distinct sub-tasks: retrieving the correct answer for a given clue (sub-task 1) and filling complete crossword grids (sub-task 2). In this work, we present

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 - 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ a.porcelli34@studenti.uniba.it (A. Porcelli); f.digravina2@studenti.uniba.it (F. D. Gravina); e.fontana7@studenti.uniba.it (E. Fontana); m.curri8@studenti.uniba.it (M. Curri); f.digregorio16@studenti.uniba.it (F. D. D. Gregorio)

🌐 mattiacurri.github.io (M. Curri)

🆔 0009-0008-7400-5820 (E. Fontana); 0009-0001-0546-7676 (M. Curri)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

our contribution to this challenge, choosing to explore the strategy of explicit length conditioning. The paper is structured as follows: Section 2 covers related work, followed by a description of the challenge sub-tasks in Section 3. The proposed approach and experimental framework are detailed in Sections 4 and 5. Main findings are discussed in Section 6, with an ablation analysis presented in Section 7. Conclusion, limitation and future work are reported in Section 8.

2. Related Work

The field of automatic crossword puzzle resolution was pioneered by Proverb [14], the first system designed to reach human-like performance using domain-specific expert modules and a large database of clue-answer pairs. Following Proverb, WebCrow [8] was developed as the first solver for non-English crosswords, specifically targeting the Italian language by utilizing the Web as its primary knowledge base. WebCrow represents the solving process as a Probabilistic Constraint Satisfaction Problem (PCSP), with a new web module which searches the web for snippets that may contain answers [15]. Another landmark system, Dr.Fill [16], treats crossword filling as a Weighted-CSP, where constraint violations are weighted and tolerated. Barlacchi et al. [17] apply a BM25 retrieval model to generate clue lists similar to the query clue from historical clue-answer database, where the generated clues get further refined through application of re-ranking models. Severyn et al. [18] introduce a distributional neural network to compute similarities between clues trained over a large scale dataset of clues that they introduce. Barlacchi et al. [3] presented SACRY, a syntax-based automatic crossword resolution system that extends the WebCrow architecture by integrating structural kernels and learning-to-rank models for effective clue reranking and answer aggregation. Rozner et al. [10] demonstrates that while a standard T5 baseline struggles at solving crosswords, a curriculum learning strategy involving synthetic wordplay tasks significantly improves performance, though the model remains unable to fully generalize compared to human experts. Wallace et al. [9] introduced the Berkeley Crossword Solver, a state-of-the-art system that integrates neural bi-encoder QA models with loopy belief propagation and local search. This approach achieved 82% perfect puzzle accuracy on New York Times crosswords, significantly outperforming Dr.Fill and top human competitors. Ciaccio et al. [4] recently proposed a contrastive learning framework for Italian crosswords, using an Asymmetric Dual Encoder architecture to model a shared latent space between clues and their solutions, significantly outperforming standard retrieval baselines in the Italian language.

3. Task Description

Sub-task 1 Given a set of clues $C = \{c_i\}_{i=1}^n$ and the expected answer length $\ell_i = |s_i|$ for each clue, the goal is to produce, for every c_i , a ranked list of candidate solutions $\hat{S}_i = (\hat{s}_{i,1}, \dots, \hat{s}_{i,m})$ such that each candidate satisfies the length constraint $|\hat{s}_{i,j}| = \ell_i$.

Sub-task 2 Given a set of crossword instances $\mathcal{G} = \{G_t\}_{t=1}^k$, each instance G_t consists of a rectangular layout (a matrix of cells) where each cell is either blocked (black) or fillable, and a set of slots V_t described by a clue and by its placement metadata: starting coordinates (x_v, y_v) and direction $d_v \in \{\text{ACROSS}, \text{DOWN}\}$. The objective is to assign a word w_v to every slot $v \in V_t$ such that $|w_v|$ matches the slot length induced by the layout, and all crossing constraints are satisfied, i.e., letters in intersecting slots agree on their shared cells.

4. Proposed Approach

Sub-task 1

The adopted model is *it5-efficient-small-el32*¹ [19], an Italian-trained variant of T5 [20] based on the Efficient T5 architecture [21]. Since matching the exact length of the ground truth is a primary challenge in crossword solving, we introduce a *Dual-Constraint* mechanism to explicitly bounds the generation process. Let $\mathcal{C}$ denote the clue and S the target solution of length L . We construct the input sequence X by embedding the target length alongside the clue:

$$X = \text{indovinello:} \oplus \mathcal{C} \oplus [\text{SL}=L] \quad (1)$$

Here, $\oplus$ represents string concatenation. The token ' $[\text{SL}=L]$ ' (Start Length) serves as an explicit prompt, instructing the model on the required character count before generation begins. Similarly, we modify the target output sequence Y to include a termination token ' $[\text{EL}=L]$ ' (End Length):

$$Y = [\text{SL}=L] \oplus S \oplus [\text{EL}=L] \quad (2)$$

For every length encountered in the training set, two tokens (one for each type) are added directly to the model's tokenizer, and their embeddings are initialized using a normal distribution. This technique acts effectively as a structural "zipper": the generation is anchored between a specific start token and a specific end token, forcing the model to plan the word construction within fixed boundaries. As an example, consider the clue "Capital of Italy" with a required solution length of 4; the input provided to the model is `indovinello: Capital of Italy [SL=4]`, and the target output on which it is trained is `[SL=4] Rome [EL=4]`. This approach differs fundamentally from simply filtering outputs during inference (e.g., masking logits to forbid the EOS token until length L is reached). If a model is not trained to anticipate the end of a word, forcing it to continue or stop artificially often leads to coherent but incorrect completions. By training with these tokens, the model learns to internalize the length constraint: the probability distribution of the generated tokens naturally converges towards the end-token ' $[\text{EL}=L]$ ' exactly when the word is complete. Evaluating whether the model actually benefits from learning the placement of the final token is left as a future ablation study.

Sub-task 2

We formulate the sub-task as a Constraint Satisfaction Problem (CSP), formally defined by the tuple $\langle V, D, \mathcal{C} \rangle$.

- V (Variables): The set of all empty slots (definitions) in the grid.
- D (Domains): For each slot $v \in V$, the domain D_v is the ranked list of candidate words composed of the top- K predictions generated by our fine-tuned model (Sub-task 1), ranked by perplexity (lower is better), and a set of K' auxiliary words retrieved from a standard dictionary². These are used only if valid model candidates are exhausted and are strictly filtered to match existing letters in the grid.
- $\mathcal{C}$ (Constraints): The set of intersection rules ensuring that horizontal and vertical words share the same letter at crossing points.

Crucially, we set $K' \ll K$ to ensure that the system behaves as a neural solver, relying on semantic understanding rather than brute-force dictionary lookups. The solving algorithm proceeds in two stages:

1. **Initialization (Seeding):** Sorting answers by perplexity we start from different positions i.e. we sets the two best words predicted by the model for the clues 1,3 as anchors. We generate multiple seed configurations, where each seed fixes a high-confidence model prediction into the grid to serve as an anchor.

¹<https://huggingface.co/gsarti/it5-efficient-small-el32>

²https://github.com/napolux/paroleitaliane/blob/main/paroleitaliane/parole_uniche.txt

2. **Recursive Backtracking:** From each seed, we perform a Depth-First Search (DFS). At each node, the algorithm attempts to fill the next slot by choosing the most probable compatible word from D_v . If a conflict arises later, it backtracks to try the next candidate.

To balance solution quality with computational efficiency, we employ an adaptive resource allocation strategy. Larger grids induce a larger search space; therefore, we scale the number of candidates (K , K') and the number of initial seeds based on the grid dimension n . The number of candidates drawn from dictionary are limited, to avoid the degeneration to a pure dictionary-based CSP solver, which would not follow the challenge constraints. The specific schedule is detailed in Table 1.

Table 1

Adaptive hyperparameters based on grid size n . K denotes model candidates, K' dictionary candidates.

Grid Size (n)	Initial Seeds	Model (K)	Dict (K')
Small ($n \leq 6$)	100	100	3
Medium ($7 \leq n \leq 8$)	500	500	5
Large ($9 \leq n \leq 10$)	1000	600	10
Extra Large ($n > 10$)	2000	600	10

Here we discuss the complexity of the algorithm. Let $m = |V|$ be the number of clues (slots) to be filled in the crossword. For each clues, the algorithm disposes of two sets of candidates: at most K words originating from the model output and at most K' words from the dictionary. The resolution proceeds via a Depth-First Search (DFS) with backtracking, which iteratively assigns an answer to a clue and incrementally verifies compatibility constraints.

Time Complexity. The computational cost associated with a single node in the search tree, which represents a specific partial assignment of a candidate word to a target slot, is dominated by the consistency verification. This operation checks the intersection constraints against the previously assigned words in the grid. Since verifying a candidate requires checking character conflicts with at most m intersecting slots (where m is the total number of clues), this step requires polynomial time and is upper-bounded by $O(m)$. In the worst-case scenario, assuming no pruning occurs, the algorithm explores up to $K + K'$ alternatives for each of the m slots. This results in a search tree with a maximum branching factor of $K + K'$ and a depth equal to m . Consequently, the total number of explorable nodes is of the order $O((K + K')^m)$. Combining the node count with the verification cost, the worst-case time complexity is $O(m \cdot (K + K')^m)$. In dominant asymptotic notation, this simplifies to $O((K + K')^m)$. In practice, to manage this complexity, we enforce the budgets described in Table 1. Table 2 reports the average runtime observed in our experiments per grid size. It is important to note that these timings refer strictly to the solving phase; they exclude the candidate generation preprocessing (which required approximately 10 minutes to generate 600 candidates using a beam search of width 1000).

Table 2

Average runtime (in seconds) for grid solving by size, using 600 candidates. Experiments were conducted on a Ryzen 7 7700 CPU. Timings exclude the candidate generation preprocessing phase.

Grid Size	Min (s)	Max (s)	Mean (s)
5×5	0.21	4.18	1.75
7×7	4.01	26.70	13.01
9×9	14.17	55.29	30.05
11×11	63.92	155.83	102.03
13×13	151.09	754.89	309.99

Space Complexity. The space complexity is $O(m)$, determined primarily by the recursion stack used by the backtracking algorithm and the data structure storing the current partial assignment (i.e.,

the mapping of words to slots). Since the algorithm operates depth-first, the maximum depth of the recursion stack corresponds exactly to the total number of clues m to be filled.

5. Experimental Setup

In this section, we detail the dataset, evaluation metrics, baselines, training configuration, and the specific settings used to assess our methodology.

Dataset Description and Data Augmentation

The dataset proposed by the challenge organizers relies on the ItaCW crossword dataset [22], significantly enriched with a collection of additional clue-solution pairs retrieved from the web. As summarized in Table 3, the data is organized to support both sub-tasks. For the first sub-task, the overall dataset comprises approximately 410,000 clue-answer pairs. These data are partitioned according to a standard scheme: 90% for the training set, 5% for validation, and 5% for testing. Regarding the second sub-task, crossword grids were generated by employing a constraint-driven, search-based construction algorithm. This algorithm was designed to populate predefined crossword layouts using exclusively valid words from the list of answers contained in the aforementioned data splits. The generation process yielded crosswords of varying sizes, resulting in 500 grids for the training set, 50 for the validation set, and 50 for the test set. To improve the vocabulary coverage, we integrate a dictionary dataset³ $\mathcal{D} = \{(d_i, w_i)\}$, containing approximately 313,000 definition-word pairs. These pairs undergo the same preprocessing (Equation 1 and 2) as the crossword clues. This augmentation could allow the model to learn the association between semantic descriptions and structural constraints on a much wider scale, generalizing its ability to define words of specific lengths beyond the stylistic syntax typical of crossword puzzles.

Table 3

Overview of the Challenge Dataset and Sub-tasks

Feature	Description/Value
Sub-task 1: Clue-Solution Pairs	
Total Size	$\approx 410,000$ pairs
Data Splits	Training (90%), Validation (5%), Test (5%)
Clue Categories	Wordplays, cryptic clues, named entities initials, fill-the-fill, etc.
Sub-task 2: Crossword Grids	
Generation Method	Constraint-driven, search-based construction algorithm
Training Set	500 grids
Validation Set	50 grids
Test Set	50 grids

Evaluation

To assess performance, we employ the official challenge metrics for each sub-task:

Sub-task 1

- **Accuracy@k** (with $k = 1, 10$): The accuracy in retrieving the correct solution word given the clue, considering the top- k system candidates.
- **Mean Reciprocal Rank (MRR)**: The average of the reciprocal ranks of the first relevant item across all clues.

³<https://huggingface.co/datasets/mik3ml/italian-dictionary>

To isolate the impact of special tokens described in Section 4, we use two modes:

1. *Constrained*: Output is filtered to include only answers matching the specified length.
2. *Unconstrained*: All generated answers are considered regardless of length.

Sub-task 2

- **Character Accuracy**: The percentage of correct characters inserted in the correct slots.
- **Word Accuracy**: The percentage of correct words inserted in the correct slots.
- **Full Grid Accuracy**: The percentage of completely solved grids.

We evaluate the algorithm by varying the number of generated candidates and testing models trained for different epoch counts. For both sub-tasks, we assess the impact of external knowledge by enabling or disabling dictionary augmentation to determine if it improves final performance.

Baselines

We compare our approach against the standard baselines provided for the tasks.

Sub-task 1 The problem is treated as an Information Retrieval (IR) task. Given a test clue $c_i \in C_{test}$, the system computes the BM25 similarity score against all training clues C_{train} . The top ten most similar clues are selected, and their corresponding answers are extracted as candidates.

Sub-task 2 This baseline combines the aforementioned ranker with a module that treats crossword puzzles as a weighted Max-SMT problem [15]. It optimizes for a solution by defining hard constraints (the grid structure) and soft constraints (candidate ranking preferences). Specifically, each clue corresponds to a disjunctive group of grid variables constrained to match candidate answers, combined conjunctively across the grid. The formulation uses the Z3 optimizer⁴ [23] with 10 candidates per clue.

Training Configuration

We trained our model using the hyperparameters outlined in Table 4. Due to time and resource constraints, we did not perform extensive hyperparameter optimization.

Table 4

Training hyperparameters and configuration details.

Hyperparameter	Value
Max Input Length	128
Max Target Length	32
Batch Size	32
Gradient Accumulation	None
Warmup Steps	100
Epochs	22–23
Learning Rate	2e-4

6. Results and Discussion

We present a comprehensive analysis of the experimental results obtained for both subtasks.

⁴<https://github.com/pncnmp/Crossword-Solver>

Table 5

Sub-task 1 results ordered by MRR. **Bold** indicates best result per metric, underlined indicates second best.

Model	Epochs	Dictionary dataset	Length tokens	Constrained?	Acc@1	Acc@10	MRR
†		-	-	-	0.69	0.83	0.72
it5-efficient-small-el32*	23	✓	✓	✓	0.58	<u>0.75</u>	<u>0.63</u>
†		-	-	-	<u>0.59</u>	0.71	0.62
it5-efficient-small-el32*	22	✓	✓	✓	0.57	<u>0.75</u>	0.62
it5-efficient-small-el32*	23	✓	✓	✗	0.55	0.72	0.60
it5-efficient-small-el32*	22	✓	✓	✗	0.54	0.73	0.60
†		-	-	-	0.51	0.73	0.57
†		-	-	-	0.47	0.67	0.53
†		-	-	-	0.46	0.69	0.52
†		-	-	-	0.43	0.59	0.47
Baseline		-	-	-	0.40	0.62	0.46
†		-	-	-	0.36	0.54	0.41

*Submitted score for the challenge

†Other participants' submission

Sub-task 1 Table 5 reports the performance for Sub-task 1, ranked by Mean Reciprocal Rank (MRR). The proposed approach significantly outperforms the challenge baseline across all metrics. Our best-performing configuration finetuned for 23 epochs with dictionary augmentation and special length tokens achieved an Accuracy@1 of 0.58 and an MRR of 0.63. This represents a substantial margin over the baseline (MRR 0.46) and confirms the efficacy of the it5 architecture for this specific domain with the addition of the special tokens. A crucial finding from Sub-task 1 is the interaction between training-time constraints and inference-time filtering. While strictly enforcing answer length during inference yields a consistent performance boost (approximately +0.03 MRR across configurations), the performance gap between models trained with special tokens and those without them is far larger. This empirical evidence suggests that the explicit tokens ($[SL = L]$ and $[EL = L]$) allow the model to internalize the length constraint, resolving semantic ambiguities and distinguishing between synonyms of different lengths during the generation process itself.

Sub-task 2 Table 6 details the solving performance on complete crossword grids. Our CSP approach demonstrates a decisive advantage over the baseline, achieving a Full Grid Accuracy of 34% compared to the baseline's 8%. An interesting divergence emerges when comparing Sub-task 1 and Sub-task 2 metrics. While the model trained for 23 epochs exhibited superior performance on isolated clues (Sub-task 1), the model trained for 22 epochs proved more effective for full grid solving (34% vs. 28% full grid accuracy). This counter-intuitive result implies that the model trained for 23 epochs may have begun to overfit. Furthermore, integrating the dictionary as a secondary candidate source consistently improved the final solution rate. Although the model successfully handles the vast majority of semantic clues (reflected in the high Character Accuracy of $\sim 92\%$), the dictionary acts as a critical fallback leading to 1 more full grid solved.

7. Ablation Study

We choose to compare both Sub-task 1 and Sub-task 2 with the best performing run submitted to the challenge.

Sub-task 1 Table 7 presents the breakdown of Sub-task 1 performance. The removal of special length tokens results in a dramatic drop in performance, with MRR falling from 0.63 to 0.43 in the absence of training-time constraints. This confirms that the standard architecture struggles to adhere to

Table 6

Sub-task 2 results ordered by Full Match Accuracy. **Bold** indicates best result per metric, underlined indicates second best.

Algorithm	Epoch	Num Candidates	Dictionary?	Char Accuracy	Word Accuracy	Full Grid Accuracy
Ours [*]	22	600	✓	<u>0.92</u>	0.85	0.34
Ours [*]	22	600	✗	<u>0.92</u>	0.85	<u>0.32</u>
Ours [*]	23	600	✓	0.93	<u>0.84</u>	0.28
Ours [*]	23	600	✗	0.91	0.82	0.22
†				0.82	0.66	0.16
†				0.82	0.67	0.16
Baseline				0.73	0.58	0.08

^{*} Submitted score for the challenge

[†] Other participants' submission

Table 7

Sub-task 1 ablation study results

Model	Dictionary dataset	Length tokens	Constrained?	Acc@1	Δ Acc@1	Acc@10	Δ Acc@10	MRR	Δ MRR
Best Performer	✓	✓	✓	0.58		0.75		0.63	
	✓	✓	✗	0.55	↓ 0.03	0.72	↓ 0.03	0.60	↓ 0.03
	✗	✓	✓	0.57	↓ 0.01	0.75	- 0.00	0.62	↓ 0.01
	✗	✓	✗	0.54	↓ 0.04	0.72	↓ 0.03	0.59	↓ 0.04
	✓	✗	✓	0.48	↓ 0.10	0.64	↓ 0.11	0.52	↓ 0.11
	✓	✗	✗	0.35	↓ 0.23	0.54	↓ 0.21	0.41	↓ 0.22
	✗	✗	✓	0.38	↓ 0.20	0.54	↓ 0.21	0.43	↓ 0.20
	✗	✗	✗	0.26	↓ 0.32	0.43	↓ 0.32	0.32	↓ 0.31

Table 8

Class Distribution Analysis of the Dataset after labeling

Category	Count	Percentage (%)
Single Class Counts		
Definition	18,665	89.65
Wordplay	1,961	9.42
Filling the fill	611	2.93
Cryptic	589	2.83
Class Combinations		
Definition	17,822	85.60
Wordplay	1,564	7.51
Filling the fill	389	1.87
Cryptic + Definition	372	1.79
Definition + Wordplay	238	1.14
Definition + Filling the fill	205	0.98
Cryptic + Wordplay	117	0.56
Cryptic	69	0.33
Cryptic + Definition + Wordplay	28	0.13
Filling the fill + Wordplay	14	0.07
Cryptic + Filling the fill	3	0.01

strict character-count constraints without conditioning the model at training time. Interestingly, while applying inference-time constraints to a model trained *without* tokens provides a significant recovery, this "post-hoc" correction remains inferior to the "intrinsic" learning provided by the tokens (MRR 0.60 without filtering). The combination of both mechanisms yields the state-of-the-art result, indicating that they are complementary: the tokens guide the semantic search, while the filter acts as a final safeguard

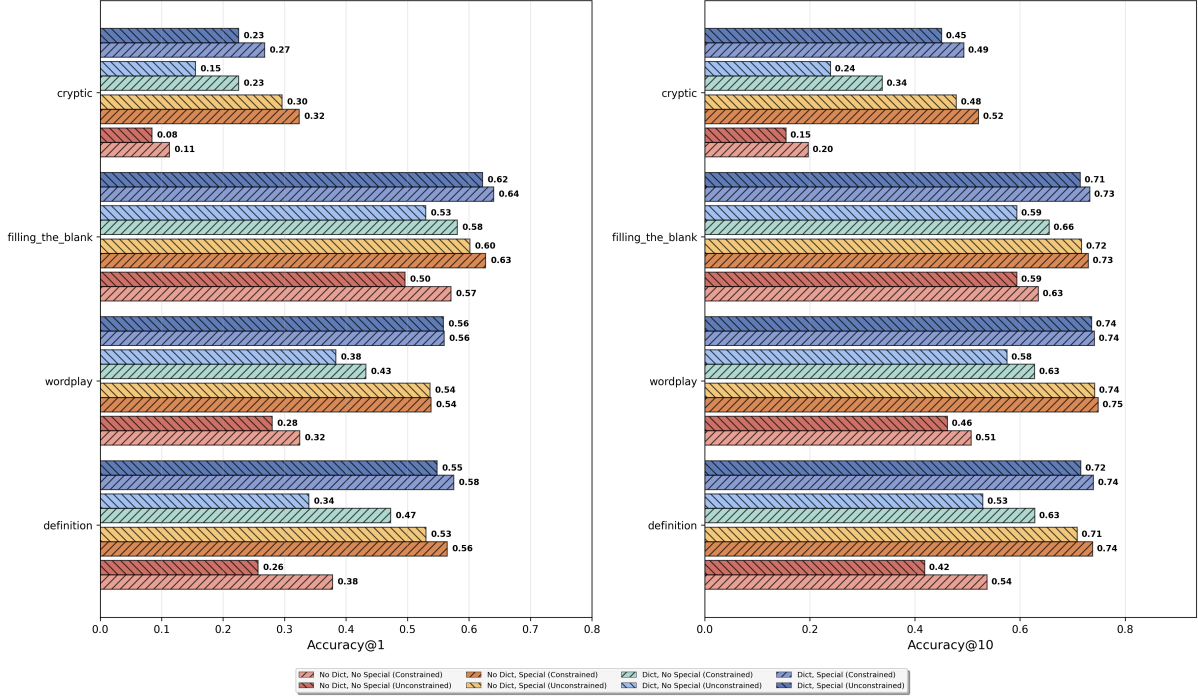


Figure 1: Sub-task 1 classification results

against generation errors. The addition of the dictionary dataset yields only marginal improvements for Sub-task 1 (e.g., +0.01 MRR). To investigate this counterintuitive phenomenon, we labeled the test set clues using Gemini 3 Flash (configured with a temperature of 0.0 and otherwise default parameters) into four distinct labels: *definition*, *wordplay*, *fill-in-the-blank*, and *cryptic*. Table 8 presents the distribution of these labels across the dataset, while Figure 1 illustrates the models’ performance across the different clue types. We observe that incorporating the dictionary and special tokens during training yields a negligible improvement in the definition category. Moreover, the gains observed in the fill-in-the-blank and wordplay classes are outweighed by a degradation in performance within the cryptic category. This analysis demonstrates that the original dataset is sufficient for capturing definitions, whereas the inclusion of a dictionary proves detrimental to the model’s ability to resolve cryptic clues. Consequently, the dictionary appears to introduce an unnecessary bias toward definitional clues. Therefore, the primary bottleneck in performance stems from complex linguistic reasoning rather than a lack of lexical coverage. However, while this analysis provides valuable insights, these conclusions must be interpreted in light of the inherent limitations and potential biases introduced by the LLM-based generation of the labels. Notably, the LLM occasionally misclassifies fill-in-the-blank clues, a limitation that reinforces some of the findings presented in our work. We release the labeled dataset in our repository, along with the adopted prompt and code to generate it.

Sub-task 2 We analyzed the behavior of the solver by varying the number of candidates k retrieved per clue, ranging from 1 to 600, as illustrated in Figure 2. The candidates are generated from the model trained for 22 epochs, using the same number of beams (1000), that is, our best performer. We observe that beyond $k = 200$, the curve begins to plateau (except for the full match accuracy), indicating that if the correct answer is not within the top 200 predictions, it is unlikely to be retrieved by simply expanding the beam further. This can be due to the fact that the finetuned model is already able to provide good candidates in the first 100-200 candidates. The ablation also highlights the “long tail” effect of the dictionary augmentation: the presence of dictionary candidates consistently maintains a performance gap over the no-dictionary setting. This proves that, while the model handles most of the grid construction, the dictionary candidates are essential for completing the final most difficult

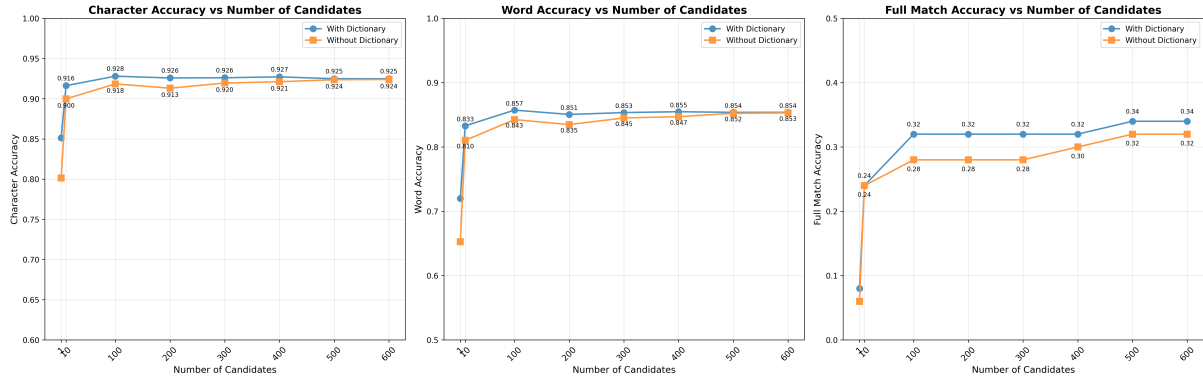


Figure 2: Sub-task 2 ablation study results

intersections, providing the necessary coverage to close the loop on complex grids.

8. Conclusion

In this work, we presented our approach to the Cruciverb-IT shared task at EVALITA 2026, addressing both crossword clue answering and complete grid solving for Italian crosswords. Our methodology combines a fine-tuned sequence-to-sequence model with architectural innovations and a constraint satisfaction problem solver. For Sub-task 1, we introduced a dual-constraint mechanism based on special length tokens ' [SL=L] ' and ' [EL=L] ' that explicitly encode answer length requirements directly into the model’s input and output sequences. Our ablation study demonstrates that this token-based approach yields substantial performance gains. The model achieved up to 0.58 Acc@1, 0.75 Acc@10 and 0.63 MRR, substantially outperforming the baseline. For Sub-task 2, we formalized crossword solving as a CSP with adaptive hyperparameters. Our backtracking solver prioritizes model-generated candidates over dictionary fallbacks and employs strict computational budgets to balance solution quality with practical runtime constraints. The solver achieved up to 0.93 character accuracy, 0.85 word accuracy, and 0.34 full grid accuracy on the test set, substantially outperforming the baseline.

Limitations and Future Work While our approach demonstrates strong performance, several directions warrant further investigation. Byte-level tokenization based models such as ByT5 [24], Charformer [25], or MambaByte [26] could provide a viable alternative to our token-based approach by natively operating at character granularity, though this would require pre-training or adapting such architectures for Italian. Remaining within the immediate scope of our proposed system, scaling our methodology to large language models (LLMs) could reveal whether the benefits of explicit structural tokens persist at greater model capacities or whether emergent abilities render such mechanisms redundant, as well as a deeper investigation into the precise impact and dynamics of the added special tokens could yield valuable insights. Beyond architectural considerations, more fine-grained hyperparameter tuning for Sub-task 2 could yield additional performance gains. Furthermore, given that the dataset for Sub-task 2 was generated using clues from Sub-task 1, future evaluations should extend to crosswords created via alternative generation pipelines. Finally, incorporating the crossword grid structure directly into the training objective for Sub-task 1 (e.g., through multi-task learning or synthetic grid completion tasks) could better align the model’s predictions with the grid’s requirements.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini 3 Pro in order to: Paraphrase and reword, Improve writing style, and Grammar and spelling check. After using these tool(s)/service(s), the

author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, M. Nissim, Cruciverb-it @ evalita 2026: Overview of the crossword solving in italian task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] G. Barlacchi, M. Nicosia, A. Moschitti, Sacry: Syntax-based automatic crossword puzzle resolution system, in: Proceedings of 53rd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Beijing, China, July. Association for Computational Linguistics, 2015.
- [4] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, Crossword space: Latent manifold learning for italian crosswords and beyond, in: Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), 2025, pp. 245–255.
- [5] G. Sarti, T. Caselli, A. Bisazza, M. Nissim, EurekaRebus - verbalized rebus solving with LLMs: A CALAMITA challenge, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 1202–1208. URL: <https://aclanthology.org/2024.clicit-1.132/>.
- [6] G. Sarti, T. Caselli, M. Nissim, A. Bisazza, Non verbis, sed rebus: Large language models are weak solvers of Italian rebuses, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 888–897. URL: <https://aclanthology.org/2024.clicit-1.96/>.
- [7] E. Mensa, L. Zane, C. J. Scozzaro, M. Delsanto, T. Milani, D. P. Radicioni, Easy to complete, hard to choose: Investigating llm performance on the proverbit benchmark, in: Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), 2025, pp. 722–734.
- [8] M. Ernandes, G. Angelini, M. Gori, et al., Webcrow: A web-based system for crossword solving, in: AAAI, 2005, pp. 1412–1417.
- [9] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, D. Klein, Automated crossword solving, arXiv preprint arXiv:2205.09665 (2022).
- [10] J. Rozner, C. Potts, K. Mahowald, Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for nlp, Advances in Neural Information Processing Systems 34 (2021) 11409–11421.
- [11] A. Sadallah, D. Kotova, E. Kochmar, What makes cryptic crosswords challenging for llms?, in: Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 5102–5114.
- [12] X. Zhang, J. Cao, C. You, Counting ability of large language models and impact of tokenization, 2024. URL: <https://arxiv.org/abs/2410.19730>. arXiv:2410.19730.
- [13] S. Saha, S. Chakraborty, S. Saha, U. Garain, Language models are crossword solvers, in: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 2074–2090.
- [14] N. Shazeer, M. Littman, G. Keim, Solving crossword puzzles as probabilistic constraint satisfaction, Proc. AAAI '99 (1999).
- [15] S. Kulshreshtha, O. Kovaleva, N. Shivagunde, A. Rumshisky, Down and across: Introducing crossword-solving as a new NLP benchmark, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume

- 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2648–2659. URL: <https://aclanthology.org/2022.acl-long.189/>. doi:10.18653/v1/2022.acl-long.189.
- [16] M. L. Ginsberg, Dr. fill: Crosswords and an implemented solver for singly weighted csps, *Journal of Artificial Intelligence Research* 42 (2011) 851–886.
 - [17] G. Barlacchi, M. Nicosia, A. Moschitti, Learning to rank answer candidates for automatic resolution of crossword puzzles, in: *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2014, pp. 39–48.
 - [18] A. Severyn, M. Nicosia, G. Barlacchi, A. Moschitti, Distributional neural networks for automatic resolution of crossword puzzles, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 199–204.
 - [19] G. Sarti, M. Nissim, IT5: Text-to-text pretraining for Italian language understanding and generation, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, ELRA and ICCL, Torino, Italia, 2024, pp. 9422–9433. URL: <https://aclanthology.org/2024.lrec-main.823>.
 - [20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, *Journal of machine learning research* 21 (2020) 1–67.
 - [21] Y. Tay, M. Dehghani, J. Rao, W. Fedus, S. Abnar, H. W. Chung, S. Narang, D. Yogatama, A. Vaswani, D. Metzler, Scale efficiently: Insights from pre-training and fine-tuning transformers, *arXiv preprint arXiv:2109.10686* (2021).
 - [22] K. Zeinalipour, T. Iaquina, A. Zanollo, G. Angelini, L. Rigutini, M. Maggini, M. Gori, Italian crossword generator: Enhancing education through interactive word puzzles (2023).
 - [23] L. De Moura, N. Bjørner, Z3: An efficient smt solver, in: *International conference on Tools and Algorithms for the Construction and Analysis of Systems*, Springer, 2008, pp. 337–340.
 - [24] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, C. Raffel, Byt5: Towards a token-free future with pre-trained byte-to-byte models, *Transactions of the Association for Computational Linguistics* 10 (2022) 291–306.
 - [25] Y. Tay, V. Q. Tran, S. Ruder, J. Gupta, H. W. Chung, D. Bahri, Z. Qin, S. Baumgartner, C. Yu, D. Metzler, Charformer: Fast character transformers via gradient-based subword tokenization, *arXiv preprint arXiv:2106.12672* (2021).
 - [26] J. Wang, T. Gangavarapu, J. N. Yan, A. M. Rush, Mambabyte: Token-free selective state space model, *arXiv preprint arXiv:2401.13660* (2024).

MINDS at Cruciverb-IT: Solving Italian Crossword Clues with Masked Language Models and Candidate Pooling

Flavio Giobergia

Politecnico di Torino. Turin, Italy

Abstract

Being able to solve crossword puzzles is a task that has engaged humans for more than a century, due to the non-trivial skills it requires – such as logical reasoning, general knowledge, multilingual competence, and wordplay. For these reasons, addressing crossword puzzles is also challenging for language models. The Cruciverb-IT shared task at EVALITA 2026 focuses on this problem for the Italian language. In this paper, we present a methodology for generating answers to crossword clues by leveraging a pre-trained encoder-only transformer model (specifically, BERT) fine-tuned to solve crosswords as a masked language modeling task. We show that a relatively small LM can be efficiently fine-tuned on a large corpus of clue-answer pairs. To mitigate limitations intrinsic to encoder-only architectures, we further propose a strategy that aggregates high-probability predictions to construct a pool of candidate solutions of varying lengths. Our approach achieves competitive performance – e.g., an accuracy@1 of 0.5910 – while remaining computationally efficient compared to approaches relying on substantially larger models.

Keywords

crossword solving, masked language modeling, encoder-only transformers

1. Introduction

Crossword puzzles have long served as a popular and intellectually demanding word game, requiring solvers to integrate linguistic knowledge, cultural references, and various forms of deductive and associative reasoning. For computational systems, these requirements translate into a complex inference task where lexical, semantic, and pragmatic cues must be combined to derive a concise answer – often a single word – from a short, sometimes ambiguous clue. As such, crosswords represent an appealing test scenario for evaluating language models’ ability to perform structured reasoning rather than surface-level text matching.

Early work on automatic crossword solving predominantly relied on lexical resources, similarity metrics, or rule-based strategies tailored to specific classes of clues. These approaches typically performed well when clues were literal or closely matched previously encountered phrasing, but their reliance on explicit linguistic patterns limited their capacity to generalize to more creative formulations, figurative uses, or clues involving polysemy and wordplay. With the emergence of neural language models, the task has gained renewed attention, yet even modern LMs continue to encounter difficulties: clues are terse, often require background knowledge, and the answer space is effectively unconstrained, making direct text generation non-trivial – especially for languages with rich morphology, such as Italian.

The Cruciverb-IT [1] shared task at EVALITA 2026 [2] provides a standardized evaluation framework for studying crossword solving in Italian. In this work, we address Task A, which focuses on predicting likely answers for isolated clues when the character length of the target word is known. This formulation isolates the semantic and lexical aspects of crossword solving, without requiring systems to fill complete grids or enforce inter-word constraints. At the same time, the task remains challenging: clues may be ambiguous or culturally grounded, and the correct answers are often short strings that do not directly appear in the clue text.

Our approach builds on the intuition that masked language modeling (MLM) can serve as an effective proxy for crossword clue completion. Instead of relying on retrieval or similarity-based methods, we

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 - 27, Bari, IT

✉ flavio.giobergia@polito.it (F. Giobergia)

id 0000-0001-8806-7979 (F. Giobergia)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

fine-tune a compact encoder-only transformer model (BERT) to reconstruct missing solution words from their corresponding clues. This modeling choice offers two advantages: (i) the ability to train efficiently on large collections of clue-answer pairs, and (ii) the use of contextualized representations that capture semantic nuances beyond surface similarity. To mitigate the limitations of encoder-only architectures – particularly the challenge of generating variable-length outputs – we introduce a candidate pooling mechanism based on high-probability token sequences, enabling the model to propose multiple plausible solutions of different lengths.

We evaluate our system on the official Cruciverb-IT benchmark and show that this lightweight approach achieves competitive performance while maintaining substantially lower computational requirements than large autoregressive LMs. Our results highlight the potential of encoder-only models, when coupled with targeted fine-tuning and structured candidate generation, to perform non-trivial reasoning over crossword clues in Italian.

2. Methodology

Our system approaches crossword clue answering by reframing it as a MLM task. Given a clue and the character length of the target word, we construct a templated input sequence in which the answer is represented by one or more [MASK] tokens. A pre-trained BERT model is then fine-tuned to reconstruct the masked span from the clue context. During inference, the true number of tokens required for the answer is unknown, which is a limitation of encoder-only architectures. To address this, we generate candidate answers by querying the model with multiple possible mask lengths and aggregating predictions across them.

The overall architecture consists of three stages:

1. Input construction and masking,
2. Fine-tuning as a masked language model,
3. Multi-length candidate generation and ranking.

Figure 1 summarizes the proposed pipeline. The following subsections explain each of the steps in more detail.

2.1. Input Construction and Masking

For each clue-answer pair in the training data, we build an input sequence that concatenates the clue text with an explicit indication of the answer length. We adopt the following template:

<clue text> (<answer length> lettere) : [MASK] [MASK] ... [MASK]

The number of [MASK] tokens corresponds to the number of tokens in the ground truth answer, as determined by the tokenizer. Including the character length in parentheses helps the model constrain its predictions to a plausible morphological and lexical space.

At inference time, however, the number of answer tokens is unknown: although the length of the answer, in characters, is known, there is a weak relationship between the length in tokens and the length in characters, as shown in Section 3.1.2. Because encoder-only models do not generate text autoregressively, we must supply a fixed number of [MASK] tokens per query. This limitation motivates the candidate-expansion strategy described in Section 2.3.

2.2. Fine-Tuning the Encoder-Only Model

We fine-tuned an Italian BERT model using standard masked language modeling with cross-entropy loss. Only the masked positions contribute to the loss, encouraging the model to reconstruct the answer tokens directly from the clue context. No additional masking is applied beyond the answer span, allowing the model to learn a direct mapping from clue text to its solution.

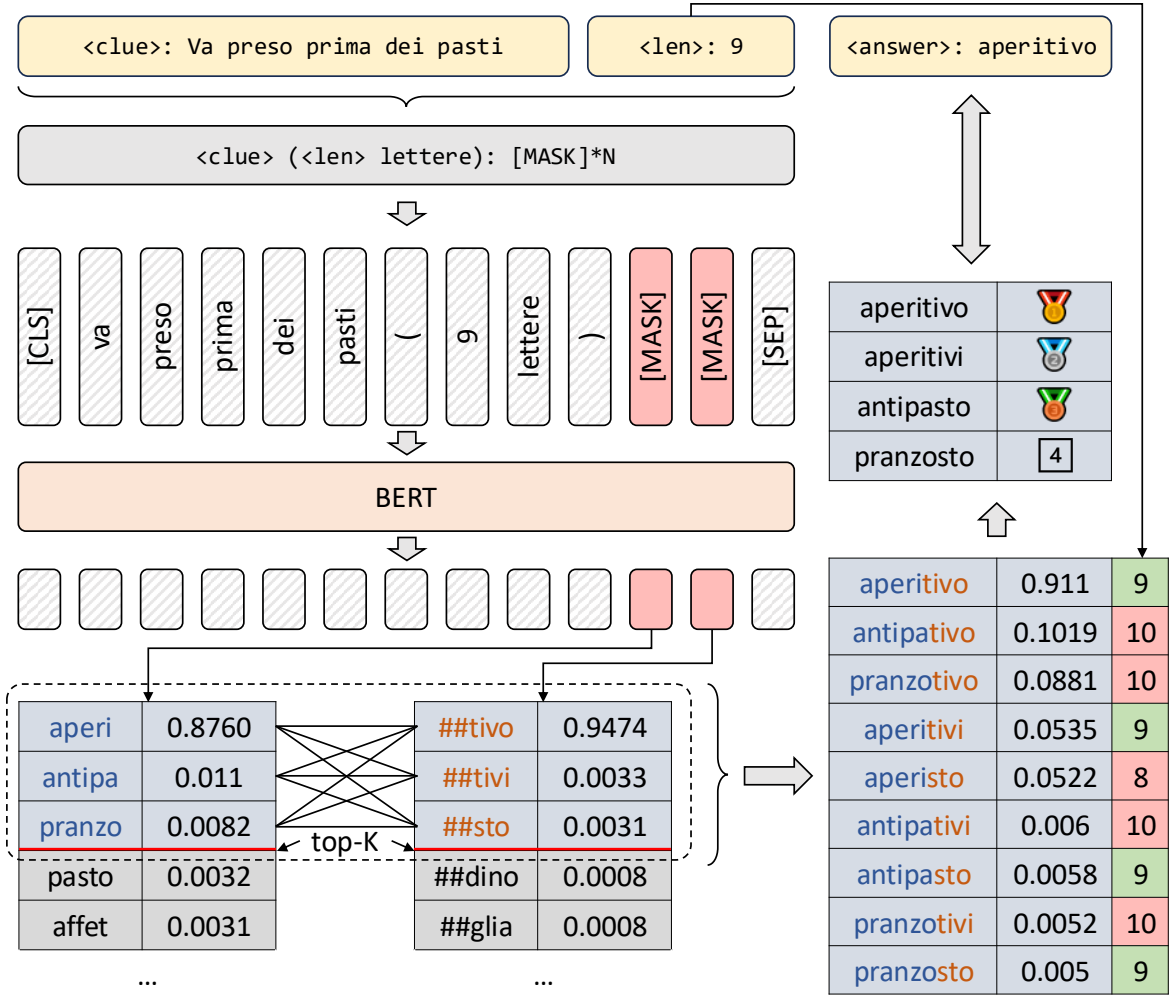


Figure 1: Summary of the proposed methodology.

2.3. Inference with Variable Mask Lengths

Since the correct number of subword tokens for a new answer is unknown, we query the model using a range of hypothesized mask lengths (e.g., 1 to N). For each length m , we create an input sequence matching the structure used during training but containing exactly m [MASK] tokens.

For each masked position, the model outputs a probability distribution over the vocabulary. Candidate answers are generated as follows:

1. **Top- K token selection per mask.** For each masked token position, we extract the K most likely predictions (according to model logits).
2. **Token-combination expansion.** For a mask span of length m , we form all possible sequences from the top- K tokens of each position, yielding up to K^m candidates.
3. **Sequence scoring.** Each candidate sequence is assigned a probability score computed as the *geometric mean* of its token probabilities:

$$P(\mathbf{t}) = \left(\prod_{i=1}^m p(t_i) \right)^{1/m}.$$

The adoption of a geometric mean allows normalizing scores across different mask lengths.

4. **Aggregation across lengths.** Candidates produced for different values of m are merged into a single ranked list. It may happen that the same word is produced at multiple mask lengths (e.g.,

aperitivo as *aperi*, *##tivo* ($m=2$) and *aper*, *##iti*, *##vo* ($m=3$)). In such cases, only the maximum-score instance is retained.

5. **Pruning invalid answers.** Only the candidate answers that are acceptable are kept. More specifically, candidates with lengths different than then known answer length, and answers containing invalid characters, are pruned – as these cannot be valid answers.
6. **Final selection.** The top 10 highest-scoring candidates are returned.

This procedure enables an encoder-only model – which cannot freely generate variable-length outputs – to approximate generative inference.

3. Experimental results

In this section, we report the results obtained on the answer generation task for Cruciverb-IT (Task 1). First, some relevant details about the dataset are presented, along with a brief overview of the metrics adopted for the evaluation (Section 3.1). Next, the main results are presented for the proposed solution (Section 3.2). The sensitivity to the two main hyperparameters of the proposed solution (N and K) is shown in Section 3.3). Finally, in Section 3.4, we conducted an error analysis to identify the most common patterns that lead to failures.

3.1. Dataset and metrics

The dataset used is the one proposed for the Cruciverb-IT challenge. It is comprised of three splits: training data (374,766 clue-answer pairs), validation data (20,820 clue-answer pairs) and test data (20,821 clue-answer pairs). Models can produce up to 10 ranked answers to a clue. The quality of models is assessed through accuracy@1, accuracy@10 and Mean Reciprocal Rank (MRR).

3.1.1. Data quality

We identify several data quality issues in the dataset:

Missing or incorrect clues/answers. A small number of entries contain missing or erroneous clues or answers, mainly due to issues in data collection. In the training set, 5 clues are reported as “<Inserire definiz.>” (“enter definition”) and 2 are entirely missing. Manual inspection also reveals a few incorrect answers, such as “*iari*” instead of “*bari*” for “*Il capoluogo della regione pugliese*”, or “*cu*” instead of “*na*” for “*Il sodio in chimica*”. These cases are rare and unlikely to significantly impact overall performance.

Same clue, different answers. Some clues appear multiple times with different answers, a common characteristic of crosswords where answer length or surrounding letters disambiguate the solution. While answer length is provided and helps rule out invalid options, the lack of contextual information makes it impossible to distinguish between multiple valid answers of the same length. In some cases, disambiguation depends on external context, such as “*personaggio del giorno*” clues, whose answers appear elsewhere in the magazine. Clues like “*Male/female name*” exhibit the highest ambiguity, with up to 17 same-length answers. Overall, the training set contains 8,685 ambiguous clue-answer pairs (about 2.3%).

Training and validation/test overlaps. After normalizing clues and answers (e.g., removing accents), we observe overlaps between the training and validation/test sets. We distinguish between *identical* overlaps (same clue and answer) and *misleading* overlaps (same clue and length, different answers). The validation set contains 255 identical and 364 misleading overlaps (about 3%), while the test set contains 269 identical and 395 misleading overlaps (about 3.2%). Identical overlaps may advantage the model, whereas misleading ones may disadvantage it.

Overall, these issues may partially affect the reported results but are expected given the dataset’s size. Using multiple evaluation metrics, including accuracy@10, helps mitigate their impact.

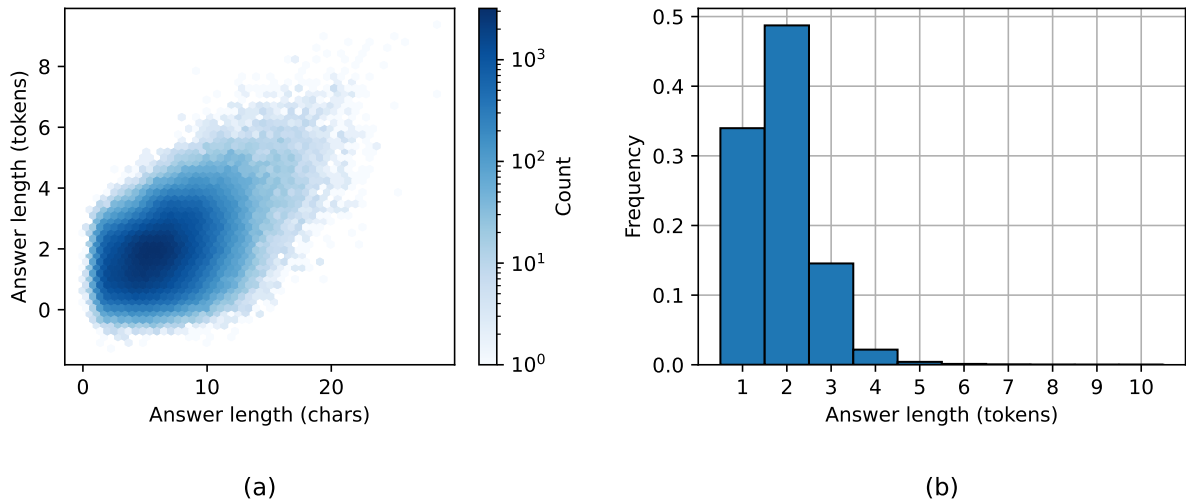


Figure 2: (a). Relationship between answer length in characters, and answer length in tokens. A clear linear relationship exists, however the variance is large enough to make it difficult to predict the “correct” number of tokens, based only on the length in characters of the answer (the only available information). **(b).** Distribution of the lengths of the answers (in tokens). The majority of answers can be represented with a limited number of tokens (97% with up to 3 tokens).

3.1.2. Answer lengths

The selected encoder-only architecture has one significant limitation: the number of tokens for the answer need to be specified in advance, as a part of the input (in the form of [MASK] tokens). The only information available about the length of the answers is in terms of letters. We qualitatively and quantitatively verify the relationship between the length of answers in tokens and in letters, to decide whether the correct number of tokens can be estimated from the length of the answer in letters.

Figure 2 (a) shows the scatter plot of lengths of answers in letters and tokens, for the training set. A subtle relationship can be observed. This relationship can be quantified with a Pearson correlation of approximately 0.51 (for the tokenizer associated with the selected version of BERT). This suggests that, although there indeed is a relationship between the two lengths, estimating the length in tokens from the length in letters may not be accurate enough.

In addition, 2 (b) shows the distribution of lengths, in tokens, for the answers in the training set. We note that 97% of the answers are composed of 1, 2, or 3 tokens. Given this limited range, we used masks of those lengths (i.e., $N = 3$) in the proposed methodology. In Section 3.3, we consider different values for N .

3.2. Main results

We report the main results obtained in Table 1. The reported performance has been obtained by training various models (as listed in the table) on the training and validation sets, and reporting the results on the test set.

We selected various BERT-based architectures, namely BERT base [3], RoBERTa [4], as well as two versions that have been pre-trained on the Italian language [5][6].

The baseline reported, presented by the organizers of the challenge, uses BM25 [7] to rank the most similar entries in the training set. The other reported solutions are the ones obtained by the other participants (the runs are unnamed as the information on the teams was not available).

The proposed solution achieves competitive results, especially considering the accuracy@1 and the MRR. In other words, if the model “knows” the solution, it ranks it in the first few positions. This is consistent with the expected behavior, since the highest probability answer is the one that has highest logits for the masked tokens. However, the gap between accuracy@1 and accuracy@10 highlights a

Table 1

Main results of the Cruciverb-IT challenge, Subtask A. For each solution, accuracy@1, accuracy@10 and Mean Reciprocal Rank are reported. For all metrics, higher is better. The named runs belong either to the proposed system, or to the baseline. All ‘XXX’ entries correspond to submissions made by other participants.

Runs	accuracy@1	accuracy@10	MRR
XXX	0.69	0.83	0.72
XXX	0.58	0.75	0.63
BERT base (Italian)	0.59	0.71	0.62
XXX	0.57	0.75	0.62
XXX	0.55	0.72	0.60
XXX	0.54	0.73	0.60
XXX	0.51	0.73	0.57
RoBERTa base (Italian)	0.5	0.64	0.54
BERT large	0.49	0.61	0.53
XXX	0.47	0.67	0.53
XXX	0.46	0.69	0.52
BERT base	0.47	0.59	0.51
XXX	0.43	0.59	0.47
Baseline	0.40	0.62	0.46
XXX	0.36	0.54	0.41

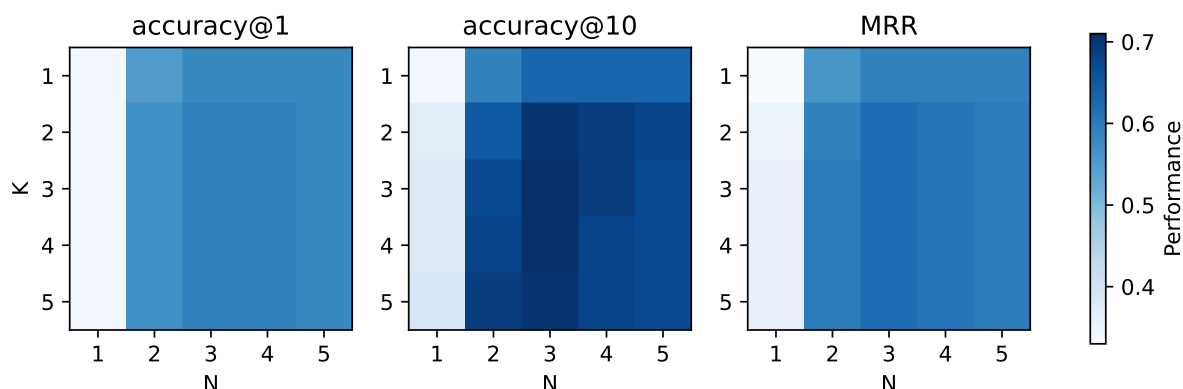


Figure 3: Performance, in terms of accuracy@1, accuracy@10 and MRR, as a function of the K and N hyperparameters.

limitation of the model: there are cases where the correct answer *could* be known (since it is in the top-10), but the model fails to rank it first. The large MRR value suggests that solutions are placed, on average, in the first few positions. However, since the model’s probabilities used are the same ones as the generating model, it is unlikely for a different answer (other than the most likely one) to be considered valid. Alternative approaches could use a different, specifically-tuned and lightweight language model (e.g., fastText [8], a fast models that supports out-of-vocabulary words [9]) to re-rank the candidates. This could allow, for instance, to reduce the probability assigned to non-existing words.

3.3. Sensitivity to hyperparameters

In this section, we focus on the sensitivity of the proposed pipeline to the two hyperparameters N (maximum number of mask tokens considered in the template) and K (when considering the K tokens with highest logits in the answer generation phase). We consider a range of values for N from 1 to 5, as this range of values covers the vast majority of answers, as previously discussed. For K , we consider values from 1 to 5 for, as we observed that considering additional values started including very low-probability tails. We report the performance as a function of K and N , according to the three metrics considered this far, in Figure 3.

The results highlight some interesting aspects, which we discuss below.

In terms of N , a value of 1 is an overall poor choice: this is expected, since the majority of answers can only be represented with 2 or more tokens (as shown in Figure 2). Indeed, using $N \geq 2$ produces more stable results.

Regarding K , various considerations can be made. First, we point out that $K = 1$ selects the highest-probability subword for each token. In other words, a single candidate answer (the “most likely” answer) is produced. This can be expected to perform poorly in terms of accuracy@10 and MRR (since both benefit from having multiple candidates being generated). We confirm this to be the case. However, this solution also has lower performance for the accuracy@1: the behavior is counterintuitive, since $K = 1$ should already provide the highest-probability answer. However, we note that the length-based pruning step may discard the only (highest probability) generated candidate, and leave the answer blank. As a consequence, having multiple candidates provides a slight boost in accuracy@1 as well.

The plots show how a choice of $N = 3$ and $K = \{3, 4\}$ produces the most promising results. We select $N = 3$ and $K = 3$, as this configuration introduces the least overhead (i.e., fewer candidates being generated and processed).

3.4. Error analysis

To better understand the failure modes of the model, we additionally present an error analysis session. We apply an existing technique, DivExplorer [10] to find “subgroups” of clues for which the proposed model systematically underperforms. We describe each clue as a bag of words, plus an additional “tag” defining the length of the expected answer.

In this way, we can extract frequent subgroups (in this case, using a support 0.001, i.e. all subgroups containing at least 0.1% of the points in the test set) and compute the performance of the model on these subgroups (in terms of accuracy@1, in this case). Then, we can extract the subgroups for which the model underperforms w.r.t. the global performance. We carry out these operations with the efficient implementation proposed in [11].

We report the 25 most diverging subgroups in Table 2. We observe some interesting families of subgroups, as highlighted in the table. Namely:

- **Long answers.** Subgroups characterized by answer lengths from 10 to 17 letters all appear in the top positions. This is intuitively a reasonable outcome: longer answers are generally more difficult, often involving multiple words (e.g., full names, or concatenations of words), leaving more room for errors.
- **General knowledge.** Another group of entries for which the model generally outperforms is the one about general knowledge. The pretrained model does possess some general knowledge, and some additional information is included during the fine-tuning phase. However, it still fails to answer many clues that rely on historical facts of general knowledge. We include “padre” (father) in this subgroup as it is typically used, both literally and figuratively (to indicate the inventor of something) in a way that still requires general knowledge to address.
- **Definitions.** Another category includes definitions: this requires knowing synonyms or being able to understand descriptions. Once again, the model does have partial knowledge of these definitions (and indeed, most of these subgroups are in the “better performing” part of the top-25). However, the model does lack some knowledge required to properly address some of the clues.
- **Play of words.** Surprisingly, there are many “play of words” clues that the model cannot solve. We argue that this is counterintuitive, as these kinds of clues (e.g., “*Il centro delle Langhe*”, or “*In the middle of Langhe*”, follow fairly simple patterns that should be easily detected by ML models. We assume, although further investigation would be required, that one of the main reasons for these kinds of mistakes lies in the fact that these models work at a token level, rather than at the character level: this has been shown in literature to cause problems when addressing character-level tasks (e.g., in the “strawberry test”) [12]. Of the various categories of errors, we believe that this is the one that can be more easily mitigated: for instance, a smaller, character-level model can be introduced to addresses these types of clues).

Table 2

Most diverging subgroups in the test set. Row colors identify four groups of patterns, namely **long answers**, **general knowledge**, **definitions**, and **play of words**. For each subgroup, the accuracy@1 for that subgroup, the performance gap with the global accuracy@1 (0.5910), and the number of hits and misses (correctly and incorrectly predicted clues) are reported.

Subgroup	accuracy@1	$\Delta_{a@1}$	# hits	# misses
<len=17>	0.0000	-0.5910	0	28
<len=15>	0.0278	-0.5632	1	35
<len=14>	0.1014	-0.4895	7	62
<len=16>	0.1351	-0.4559	5	32
<len=13>	0.1845	-0.4065	19	84
modo	0.2683	-0.3227	11	30
esserlo	0.2857	-0.3053	6	15
<len=12>	0.2869	-0.3041	70	174
far	0.2895	-0.3015	11	27
<len=11>	0.3326	-0.2584	146	293
personaggio	0.3333	-0.2577	12	24
storico	0.3333	-0.2577	10	20
protagonista	0.3333	-0.2577	7	14
italiano	0.3429	-0.2481	12	23
<len=10>	0.3592	-0.2318	315	562
lavora	0.3600	-0.2310	9	16
tedesco	0.3636	-0.2274	8	14
sistema	0.3636	-0.2274	8	14
piccola	0.3636	-0.2274	8	14
centro	0.3696	-0.2214	34	58
<len=2>, centro	0.3778	-0.2132	17	28
padre	0.3793	-0.2117	11	18
anni	0.3793	-0.2117	11	18
<len=2>, fondo	0.3810	-0.2100	8	13
parola	0.3810	-0.2100	8	13

4. Conclusions

We presented an approach to Italian crossword clue solving based on fine-tuning an encoder-only transformer as a masked language model, together with a candidate generation phase to handle variable-length answers. We overcome one of the limitations of encoder-only models by aggregating high-probability token sequences across multiple mask lengths. Experimental results show that this lightweight approach achieves competitive performance, particularly in terms of accuracy@1 and MRR, despite relying on significantly smaller models than autoregressive alternatives. Our analysis highlights both the strengths of MLM for lexical inference and its limitations in handling long answers and character-level wordplay. To address these limitations, future work could explore hybrid architectures that work at a character-level, as well as retrieval-based components.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors’ views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT (GPT 5.2) in order to: Drafting content. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell'Orletta, M. Nissim, Cruciverb-it @ evalita 2026: Overview of the crossword solving in italian task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.
- [4] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
- [5] B. Staatsbibliothek, S. Schweter, bert-base-italian-cased (revision 843e404), 2025. URL: <https://huggingface.co/dbmdz/bert-base-italian-cased>. doi:10.57967/hf/5850.
- [6] roberta-base-italian, 2025. URL: <https://huggingface.co/osiria/roberta-base-italian>.
- [7] S. Robertson, H. Zaragoza, The probabilistic relevance framework: BM25 and beyond, volume 4, Now Publishers Inc, 2009.
- [8] T. Mikolov, E. Grave, P. Bojanowski, C. Puhersch, A. Joulin, Advances in pre-training distributed word representations, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.
- [9] C. Savelli, F. Giobergia, Enhancing cross-lingual word embeddings: Aligned subword vectors for out-of-vocabulary terms in fasttext, in: 2024 IEEE 18th International Conference on Application of Information and Communication Technologies (AICT), IEEE, 2024, pp. 1–6.
- [10] E. Pastor, L. De Alfaro, E. Baralis, Looking for trouble: Analyzing classifier behavior via pattern divergence, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 1400–1412.
- [11] F. Giobergia, E. Pastor, L. de Alfaro, E. Baralis, Detecting interpretable subgroup drifts, in: Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1, KDD '25, Association for Computing Machinery, New York, NY, USA, 2025, p. 366–377. URL: <https://doi.org/10.1145/3690624.3709259>. doi:10.1145/3690624.3709259.
- [12] N. Xu, X. Ma, Llm the genius paradox: A linguistic and math expert's struggle with simple word-based counting problems, in: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 3344–3370.

UNIBA at Cruciverb-IT: Solving Italian Crosswords with Encoder–Decoder Models and Beam Search

Pierpaolo Basile¹

¹Department of Computer Science, University of Bari Aldo Moro, Bari, Italy.

Abstract

This paper describes the UNIBA system submitted to the Cruciverb-IT task at EVALITA 2026, the first shared evaluation campaign focused on solving Italian crossword puzzles. The task comprises two subtasks: (i) generating candidate answers for crossword clues and (ii) autonomously filling complete crossword grids while enforcing character-level consistency. Our approach relies on an encoder–decoder architecture based on IT5, a transformer model pretrained for Italian, and avoids the use of external resources or large proprietary language models. To better simulate realistic crossword solving, we train the model to generate answers conditioned not only on the clue and answer length, but also on partially filled solutions, obtained by masking characters in known answers. For grid completion, we propose a dynamic beam search strategy that iteratively updates candidate answers as the grid evolves. Candidate solutions are ranked using a binary classifier trained to estimate answer correctness from structural and model-based features. Experimental results show that exploiting partial solutions significantly improves clue answering and that the proposed grid-filling strategy outperforms the official baseline on all evaluation metrics, narrowing the gap with the best-performing systems. These findings confirm the effectiveness of lightweight encoder–decoder models combined with search-based strategies for crossword solving in Italian.

Keywords

Artificial Intelligence, Crossword solving, Encoder-decoder architecture, Beam search

1. Introduction and Motivations

This report describes the participation of the UNIBA team in the Cruciverb-IT task [1] organized within EVALITA 2026 [2], a periodic evaluation campaign of Natural Language Processing (NLP) and speech tools for the Italian language. Cruciverb-IT is the first shared task proposed at EVALITA on crossword puzzle solving in Italian. The task is composed of two subtasks: 1) answering clues extracted from Italian crosswords; 2) autonomously solving Italian crossword grids.

Subtask 1 involves solving clues drawn from Italian crossword puzzles. The task is framed as a question-answering problem: participants are given a set of clues ($C = c_1, c_2, \dots, c_n$) and are required to develop a system that, for a given clue (c_i), generates one or more candidate solutions ($S = s_1, s_2, \dots, s_k$), ideally including the correct answer (s_i). To better approximate a realistic crossword-solving scenario and to further constrain the search space, each clue (c_i) is accompanied by the character length of its target answer (s_i). For example, given the clue and target length “*Un passo indietro nel tempo*”, ⁴1, the system should output a list of candidate answers that ultimately includes the correct solution, “*ieri*”².

Subtask 2 focuses on the autonomous solving of Italian crossword grids. Participants are provided with a set of empty crossword grids ($G = G_1, G_2, \dots, G_k$), where each grid (G_i) is associated with a list of clues. Each clue is annotated with the $((x, y))$ coordinates of the starting cell of its corresponding answer in the grid, as well as its direction—either across (*orizzontale*) or down (*verticale*). Each crossword grid (G_i) is represented as an $(R \times R)$ matrix, in which each cell is either empty or blocked by a black square. The developed systems are required to autonomously fill the grid with suitable answers,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ pierpaolo.basile@uniba.it (P. Basile)

🌐 <https://www.uniba.it/it/docenti/pierpaolo-basile> (P. Basile)

🆔 0000-0002-0545-1105 (P. Basile)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹In English: “A step back in time”

²In English: “Yesterday”

producing a fully or partially completed crossword that enforces character-level consistency at word intersections and maximizes the number of correctly placed solutions.

A language game has already been proposed in two editions of EVALITA (2018 and 2020) [3, 4]. The game was based on the TV game “La Ghigliottina”, which involves a single player, who is given a set of five words - the clues - each linked in some way to a specific word that represents the unique solution of the game. Words are unrelated to each other, but each of them has a hidden association with the solution. Once the clues are given, the player has one minute to find the solution. The presence of other language-based tasks in previous editions of EVALITA supports the interest and vitality of the research community in this kind of task.

The Cruciverb-IT task is particularly timely, as language games have recently gained attention as effective benchmarks for probing the reasoning and interpretative capabilities of Language Models (LMs). Among them, crossword puzzles constitute a particularly demanding task, as they require not only lexical and syntactic competence but also cultural knowledge, lateral thinking, and the ability to resolve ambiguity and polysemy [5, 6, 7, 8]. Crossword solving involves complex semantic and pragmatic inference, making it a valuable setting for evaluating deeper language understanding beyond surface-level similarity matching.

Before the emergence of modern LMs, crossword-solving systems primarily relied on retrieval-based approaches and shallow lexical or semantic features [9, 10]. Subsequent work introduced more structured linguistic information, such as lexical resources, syntactic cues, and ranking strategies, to improve clue-answer matching in Italian [11, 12]. However, these approaches remain limited when addressing clues that require non-literal interpretation, wordplay, or semantic ambiguity resolution.

Despite significant progress by Large Language Models (LLMs), solving crossword puzzles remains challenging, especially for under-resourced languages such as Italian [13]. Our primary motivation is to build an effective solution without relying on closed LLMs or fine-tuning open LLMs with larger numbers of parameters. For that reason, we choose to work with an encoder-decoder architecture specific to Italian (IT5 [14]) with a small number of parameters compared to current small-to-medium LLMs. Moreover, to fill the crossword, we want to use a dynamic algorithm that updates the list of candidate solutions based on the characters already inserted in the grid. For that purpose, we train IT5, taking partial answers into account when we already know some characters. All the methodological details are reported in Section 2, while results and discussion are provided in Section 3.

2. Description of the System

Our approach is based on the idea of solving a crossword as humans do. Humans usually approach a crossword puzzle by first scanning the clues and filling in the answers they immediately recognize. These early answers provide anchor letters that help narrow down the possibilities for more difficult clues. Solvers often alternate between across and down clues, using intersecting letters to test hypotheses and eliminate incorrect options. Over time, the grid becomes increasingly constrained, allowing previously ambiguous clues to be solved through cross-checking and incremental refinement.

Generally, several approaches in the literature are based on the idea of providing a list of candidate solutions and then using it to fill the crossword using constraint programming or similar techniques. Our idea is to generate solutions based on the clue and the partial solution, if some characters are already available in the crossword. For these reasons, we train an encoder-decoder model that takes as input not only the clue but also the partial solution. For example, given the clue “*Un passo indietro nel tempo*”³ and the solution “*ieri*”⁴, we generate all the examples that involve all the possible partial solutions obtained by masking one or more characters.

The Table 2 reports a portion of all the possible partial solutions of the word “*ieri*”. Given a word composed of n characters, all the possible partial solutions are 2^n . We also add the number of missing characters in the solution and the solution length to the encoder’s input text.

³In English: “A step back in time”

⁴In English: “Yesterday”

Partial solution	Missing chars
ieri	0
i_ri	1
i__i	2
_e__	3

Table 1

Example of partial solutions of the word: “ieri”

Using this approach, we can prompt the model even when we know some of the final solution’s characters. Obviously, we also prompt the model where no characters are known. By following this idea, we built our approach. The whole pipeline is composed of several steps:

1. produce training data using the initial training set provided by the task organizers;
2. train the encoder-decoder model on training data;
3. define a strategy for selecting the best solution given the current crossword and the list of clues;
4. define a solution strategy for filling the crossword.

For step 3, we rely on the score produced by the encoder-decoder strategy during generation, while for step 4, we try different solution search algorithms, such as greedy, A*, and beam search. After a preliminary evaluation on the development set provided by the organizers, we chose beam search because it offers the best compromise between performance and execution time. In this report, we report only the details about the beam search implementation. The following subsections provide more details on each step.

2.1. Encoder-Decoder Training

Our idea is to generate for each pair of clue and answer several training examples in which some characters of the original answer are masked using the underscore character. Formally, given an answer of n characters, all the possible partial solutions that we can obtain by masking one or more characters are equal to 2^n . Following this approach, starting from the original training data of 374,766 pairs, we obtain 16,254,904 training examples. Each example is composed of: 1) the partial solution obtained by masking some characters (*partial_answer*); 2) the number of missing characters in the answer (*missing_chars*); 3) the correct answer (*answer*) and 4) the total number of characters that compose the correct answer (*answer_length*). All training data are stored in a JSON line file, as shown in Listing 1.

Listing 1: “Example of training data.”

```

1 {"partial_answer": "_di", "missing_chars": 1, "answer": "idi", "answer_length": 3
  , "clue": "Giorni di met\u00e0 mese nell'antica Roma"}
2 {"partial_answer": "_l__ari", "missing_chars": 3, "answer": "alamari", "
  answer_length": 7, "clue": "Possono sostituire le mostrine"}
3 ...

```

We train three models. The first two models are trained using the dataset composed of partial answers. The difference between the two models lies in how the input text is represented:

- **Model 1:** the input is formatted using special tokens that are added to the tokenizer in the following way:

```
[CIT_MISS_LENGTH] {0} [CIT_LENGTH] {1} [CIT_CLUE] {2} [CIT_ANSWER] {3}
```

where {0} is the number of missing characters, {1} is the answer length, {2} is the clue and {3} is the partial answer. The added special tokens are: [CIT_MISS_LENGTH], [CIT_LENGTH], [CIT_CLUE] and [CIT_ANSWER].

- **Model 2:** the input is formatted using plain text without the usage of special tokens in the following way:

```
"Trova la soluzione dove _ indica un carattere mancante. Caratteri mancanti  
: {0}. Lunghezza soluzione: {1}. Soluzione parziale {2}. Indizio: {3}"
```

where $\{0\}$ is the number of missing characters, $\{1\}$ is the answer length, $\{2\}$ is the partial answer and $\{3\}$ is the clue.

Finally, **Model 3** is trained only on the original dataset without using a partial solution. The input text is formatted as for the Model 2:

```
"Trova la soluzione. Lunghezza soluzione: {0}. Indizio: {1}"
```

where $\{0\}$ is the answer length and $\{1\}$ is the clue. For this kind of input, where partial answers are not exploited, we trained the model without special tokens, as we observed on the organizers' validation data that worse results were obtained when they were used. All the results on validation and test data are reported in Section 3.

For all three models, the decoder's output corresponds to the correct answer. All models are based on IT5 Large⁵ [15], an encoder-decoder transformer model pretrained specifically for Italian. IT5 is trained on a cleaned web-crawled Italian corpus including more than 40 billion words. We did not add further training data, such as dictionary definitions. Our idea is to build the model solely from the provided data.

For all the models, we use the following parameters: learning rate = $5e-5$, batch size = 32, weight decay = 0.01. We use different numbers of epochs: the models that exploit the complete training data with partial answers are trained on a single epoch, while model 3 is trained on 10 epochs. We use a single GPU NVIDIA RTX A6000 with 48GB of VRAM. On the validation set, Model 1 performs worse than Model 2; therefore, we exclude it.

2.2. Fill the Grid

The strategy used to fill the grid relies on previous models to produce, for each clue, a list of candidates based on the clue and the characters already present in the crossword. The list of candidates is not computed only once when the crossword is empty, but is recalculated at each step, using characters added to the crossword during the filling process. To exploit the solution space, we implement a beam search algorithm that selects the best k alternatives at each step and follows them in the next step. One main problem is choosing the best candidate answer at each step to put into the crossword. The idea is to follow the human approach: select the answer with the highest confidence. As a first approach, we rely on the score generated by the beam search of the IT5 decoder during the answer generation. We observe that this approach does not yield good results; therefore, we chose to exploit the correct grids (solved crosswords) provided by the organizers as training data.

The idea is to build a binary classifier that takes a candidate solution as input and predicts whether it is a correct answer. Several features represent the candidate solution:

1. the number of rows and columns in the grid;
2. the grid completeness that is the ratio between filled cells and the total number of cells to fill in the grid;
3. the starting row and column of the current clue where the candidate solution belongs;
4. the number of missing characters, this feature is used to track if some characters of the solution are already known;
5. the ratio between missing characters and the answer length;
6. the direction: across or down;
7. 1 if the candidate solution belongs to the list of answers observed in training data, 0 otherwise;
8. the beam search score provided by the decoder during the generation.

⁵<https://huggingface.co/gsarti/it5-large>

For each grid in the training data, we simulate a game in which clues are selected at random until the grid is completed. At each step, when a clue is selected, we generate training data for each candidate solution to the current clue. For IT5, we use a beam size of 100 at each step to produce a list of almost 100 candidates for each clue. The generated list may be less than 100 since we remove all candidates that contain spaces, violate the grid constraints or whose lengths differ from the answer length. This approach is the same as the one used during the crossword filling in the testing phase, since the answer length is a known parameter when solving the crossword. The training dataset contains 500 grids, and the number of generated examples used to train the binary classifier is 65,863. Obviously, the majority of the examples are classified as negative, since the incorrect candidates outnumber the correct ones. We use XGBoost as the model behind our classifier and, to account for the dataset’s imbalance, we set the `scale_pos_weight` parameter to the ratio of negative to positive classes. The other XGBoost parameters are set as follows: `n_estimators=5000`, `learning_rate=0.05` and `max_depth=6`.

To test the binary classifier’s performance, we compute its accuracy on the validation grids provided by the organizers. The validation contains 50 grids and generates 14,502 evaluation examples. The performances are summarized in Table 2.2. The classifier obtains satisfactory results. Nevertheless, it struggles with the positive class, which is expected given the dataset’s high imbalance. The number of false positives is 765 (5.2%), while the number of false negatives is 215 (1.5%).

class	precision	recall	f1-score	support
0	.98	.94	.96	13,084
1	.61	.85	.71	1,418
accuracy	.93	14,502		
macro avg	.80	.89	.84	14,502
weighted avg	.95	.93	.94	14,502

Table 2

Performance of the binary classifier used to classify candidate answers.

These performances can be considered acceptable, since the beam search algorithm that exploits the candidate list does not select a single answer from the list, but instead ranks candidates according to the probability of each class provided by the classifier. In particular, if the classifier predicts class 1, the score is equal to $1 + P(class_1)$; otherwise, the score is $1 - P(class_0)$.

When we try to solve the crossword, at each step we compute and rank the list of candidate answers for all unfilled words in the grid. Then, we select the unfilled word with the candidate answer that obtains the highest score and generate a beam for each candidate answer of that word to exploit more solutions. These beams are added to the list of possible beams, and only the k most promising are retained. The sum of the current score and scores of all selected answers from the previous steps ranks the beams.

When we produce the list of candidate answers for each unfilled word, we take into account several events:

- we remove all candidates that contain spaces;
- we remove all candidates that do not respect the expected length;
- we remove all candidates that do not respect the grid constraints, for example, if crossing cells are already filled with a different character.

During the candidates generation for a specific unfilled word, the generated list may be empty; in this case, a fake solution composed of spaces⁶ with a very low score ($1e - 6$) is added to the list. The spaces are not considered when the algorithm checks whether a candidate satisfies the grid constraints; this gives the algorithm greater flexibility, allowing it to proceed even when no candidate solutions are available. This means that it is possible to generate incomplete crosswords.

We produce two variants of beam search: the former exploits only candidate solutions labeled as positive by the binary classifier, whereas the latter retains all candidates.

⁶We retain crossing characters already present in the grid.

It is essential to underline that at each step of the beam search, if characters are already available in the grids, they are used to form the partial solution, which is then fed to the encoder-decoder to generate a list of possible candidates.

In all experiments, we set the encoder-decoder beam search to 100, while the grid-filling algorithm uses a beam search of 20. We select these parameters according to a preliminary evaluation on the validation set. The only parameter that we change concerns the use of candidate answers labeled as negative. In submitted runs, we exclude the model trained with special tokens (model 1) because it performs the worst.

3. Results and Discussion

In this section, we report the results obtained in each subtask.

For **subtask 1**, we perform two runs. The first run (RUN1) relies on the encoder-decoder trained on the original dataset, while the second run (RUN2) uses the model trained on partial answers. For both models, we use a beam size of 100 and retain only the first 10 answers that do not contain spaces and that are within the expected answer length, since the subtask allows a maximum of 10 solutions per clue. It is essential to note that when generating the model input in RUN2, we mask all characters in the partial solution and set the number of missing characters to the answer length.

Metrics for subtask 1 are $acc@1$ and $acc@10$, which are the accuracy in retrieving the correct solution word given the corresponding clue, considering the top 1 and 10 candidates produced by the system, and Mean Reciprocal Rank (MRR), that is, the average of the reciprocal ranks of the first relevant item across all clues. Table 3 shows the results on subtask 1. We observe that the model trained on partial answers achieves a remarkable gain over the model trained only on the complete answer. However, the best performance (RUN2) is close to the organizers’ baseline and far from the best system. We have no details about the best system’s approach, but we want to underline that we do not exploit external data, such as dictionaries or large language models, whose use is allowed.

System	acc@1	acc@10	MRR
<i>Best system</i>	0.69	0.83	0.72
UNIBA RUN2	0.43	0.59	0.47
<i>Baseline</i>	0.4	0.62	0.46
UNIBA RUN1	0.36	0.54	0.41

Table 3
Results of the subtask 1.

Regarding **subtask 2**, we provide two runs. RUN1 takes into account all the candidates, while RUN2 considers only the candidates labeled as positive by the binary classifier. Metrics for subtask 2 are: $char_acc$, which is the accuracy in inserting the correct characters in the correct slots; $word_acc$, the accuracy in inserting the correct word in the correct slots; $full_match_accuracy$, the accuracy in solving the entire grid. A partially filled grid will be evaluated, counting empty squares as errors. Results are reported in Table 3. Essentially, both runs provide similar results. It is probably because candidate answers labeled as negative receive low scores and are discarded during beam search. This happens where candidates labeled as positive exist for each unfilled word. In fact, negative answers are

System	char_acc	word_acc	full_match_accuracy
<i>Best system</i>	0.92	0.85	0.34
UNIBA RUN2	0.82	0.67	0.16
UNIBA RUN1	0.82	0.66	0.16
<i>Baseline</i>	0.73	0.58	0.08

Table 4
Results of the subtask 2.

a fallback when no positive candidates are available. In testing data, this case is probably rare, and the two runs provide similar results.

Looking at the results, we observe a clear gain over the baseline. Compared to the baseline, we achieve a remarkable improvement in performance, even though our system is very close to it on subtask 1. The baseline leverages the output of subtask 1, combined with an additional module that optimizes for a solution by maximising satisfied constraints while respecting the grid’s hard constraints. By treating crossword puzzles as a weighted Max-SMT problem, as partially described in [16]. The final problem formulation is passed to the Z3 optimizer, which satisfies all hard constraints and maximizes the weighted satisfaction of soft constraints. This means that our approach, which exploits the characters already present in the grids to build the candidate list of answers at each step, can improve overall performance, even if the accuracy when all characters are unknown is similar to the baseline.

The gap relative to the best system is also reduced, demonstrating the effectiveness of our strategy for filling the grid using the available answers. Moreover, beam search can deliver remarkable results even though its complexity is lower than the baseline, and the entire solution space is not explored, which does not guarantee finding the best solution.

It is also interesting to analyze the system’s accuracy in solving crosswords as a function of the grid’s size. Table 3 shows the accuracy of each system in solving grids of different sizes. We observe that no system can solve grids larger than 9, and UNIBA’s performance on a grid of size 9 is equal to that of the best system. However, the best system delivers very good results on small grids. The baseline provides a correct grid only for size 5.

System	5x5	7x7	9x9	11x11	13x13
<i>Best System</i>	0.9	0.6	0.1	0	0
UNIBA RUN1	0.4	0.3	0.1	0	0
UNIBA RUN2	0.4	0.3	0.1	0	0
<i>Baseline</i>	0.4	0	0	0	0

Table 5

Accuracy at different grid sizes.

Finally, our approach delivers remarkable results that support our motivations behind the designed methodology. The approach seems to be penalized by the encoder-decoder model’s performance when all the solution’s characters are unknown. This can depend essentially on two factors: the quality of the base model (IT5) and the use of only the training data provided by the organizers, which limits the model’s ability to generalize. It could be interesting to test the approach on other languages, for example, English, using the T5 family of models, or to test multilingual T5 models on Italian instead of relying on IT5. The quality of the training data can be improved by adding definitions from dictionaries, as already done in other approaches to solving Italian crosswords.

3.1. Additional Experiments

After the task submission deadline and after the test becomes available, we perform additional experiments. In particular, we aim to improve the beam search algorithm for grid filling. One limitation is that the beam search at each step selects the unfilled word with the best candidate answer, then expands the search by exploring the other candidates for that word. This approach limits the exploration of other unfilled words. For that reason, we modify the beam search and compute the following grids at each step, taking into account all candidates for all unfilled words. Then we retain only the best k beams. In this way, we try to explore more unfilled words at each step rather than focusing only on the unfilled word with the best candidate. However, this approach has a drawback: using a fixed beam width can limit the exploration of unfilled words in large grids. We overcame this limit by using a dynamic beam width. We use a fixed factor bw_f and multiply it by the number of unfilled words in the grid. Using this approach, it is possible to explore more grids at the start when more words are unfilled. Using a $bw_f = 5$, we obtain an increment of solved grids from 0.16 to 0.2 while *char_acc* and *word_acc* remain

the same (0.81 and 0.67). In particular, the number of solved 5x5 grids increases from 4 to 6, achieving an accuracy of 0.6, while the accuracy for other grid sizes remains the same. This result can be explained by the encoder-decoder’s low performance, which is a bottleneck in the search for the best solution.

Finally, we conduct an experiment to understand how performance changes as the number of missing characters in the correct answer increases. Using the validation data from subtask 1, we build an evaluation dataset that considers all possible partial solutions derived from the correct answer by removing characters. Since this dataset is very large and a complete evaluation would take a long time, we consider a random 25% subset. Results in Table 6 show that performance increases with the percentage of known characters relative to the length of the correct answer. These results prove the effectiveness of our model in finding the correct answer when some characters are already present in the crossword.

% known chars.	acc@1	acc@10	MRR
<10	0,0579	0,2828	0,1017
<20	0,0698	0,3231	0,1115
<30	0,0759	0,3488	0,1222
<40	0,0877	0,396	0,1372
<50	0,0892	0,4048	0,1418
<60	0,0989	0,4392	0,1512
<70	0,1007	0,4439	0,1525
<80	0,1159	0,5215	0,1851
<90	0,1385	0,561	0,1849

Table 6

Performance according to the percentage of known characters of the correct answers.

4. Conclusions and Future Works

In this paper, we present the UNIBA system submitted to the Cruciverb-IT task at EVALITA 2026, addressing both clue answering and crossword grid completion in Italian. Our approach combines an encoder–decoder model pretrained for Italian with a dynamic, search-based strategy that incrementally exploits partial solutions during grid filling. By explicitly modeling partially known answers and updating candidate generation as the grid evolves, the proposed system closely mimics human behavior in crossword solving.

Experimental results show that training the encoder–decoder model on partially masked answers leads to consistent improvements in clue answering performance compared to a model trained only on complete answers. More importantly, the proposed grid-filling strategy significantly outperforms the official baseline on all metrics for subtask 2, demonstrating that dynamically exploiting intersecting letters is crucial for effective crossword completion. These results are achieved without relying on external lexical resources, dictionaries, or large proprietary language models, highlighting the effectiveness of lightweight, task-specific architectures combined with structured search strategies.

At the same time, our analysis reveals several limitations. The overall performance remains strongly influenced by the quality of the candidate answers generated when no characters are initially available, which represents a bottleneck for both subtasks. Moreover, despite using beam search, solving larger grids (11×11 and above) remains challenging, suggesting that more expressive candidate generation or stronger global reasoning is required to fully explore the solution space.

Future work will focus on improving the encoder–decoder component, for instance, by augmenting the training data with high-quality external resources such as dictionary definitions or encyclopedic descriptions, which are known to be beneficial for crossword solving. Another promising direction is evaluating multilingual or larger T5-based models to assess whether improved semantic representations can mitigate the limitations observed when partial information is unavailable. From a search perspective, further refinements of the grid-filling algorithm—such as adaptive beam widths, more informed clue-selection strategies, or tighter integration between generation and constraint satisfaction—could improve

scalability to larger grids. Finally, extending the proposed framework to other languages and crossword datasets would help assess its generality and robustness beyond the Italian setting.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly to: Grammar and spelling check, Paraphrase and reword, and Improve writing style.

References

- [1] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, M. Nissim, Cruciverb-it @ evalita 2026: Overview of the crossword solving in italian task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [3] P. Basile, M. De Gemmis, L. Siciliani, G. Semeraro, Overview of the evalita 2018 solving language games (nlp4fun) task, *EVALITA Evaluation of NLP and Speech Tools for Italian 12* (2018) 75.
- [4] P. Basile, M. Lovetere, J. Monti, A. Pascucci, F. Sangati, L. Siciliani, et al., Ghigliottin-ai@ evalita2020: Evaluating artificial players for the language game “la ghigliottina”, in: *CEUR WORKSHOP PROCEEDINGS, AILC-Associazione Italiana di Linguistica Computazionale*, 2020, pp. 345–348.
- [5] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, D. Klein, Automated crossword solving, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3073–3085. URL: <https://aclanthology.org/2022.acl-long.219/>. doi:10.18653/v1/2022.acl-long.219.
- [6] J. Rozner, C. Potts, K. Mahowald, Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for nlp, *Advances in Neural Information Processing Systems 34* (2021) 11409–11421.
- [7] S. Saha, S. Chakraborty, S. Saha, U. Garain, Language models are crossword solvers, in: *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2025, pp. 2074–2090.
- [8] A. Sadallah, D. Kotova, E. Kochmar, What makes cryptic crosswords challenging for llms?, in: *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 5102–5114.
- [9] M. Ernandes, G. Angelini, M. Gori, et al., Webcrow: A web-based system for crossword solving, in: *AAAI*, 2005, pp. 1412–1417.
- [10] G. Angelini, M. Ernandes, M. Gori, Solving italian crosswords using the web, in: *Congress of the Italian Association for Artificial Intelligence*, Springer, 2005, pp. 393–405.
- [11] G. Barlacchi, M. Nicosia, A. Moschitti, A retrieval model for automatic resolution of crossword puzzles in italian language, in: *Proceedings of the First Italian Conference on Computational Linguistics CLiC-it 2014 & and of the Fourth International Workshop EVALITA 2014: 9-11 December 2014*, Pisa, Pisa University Press, 2014, pp. 33–37.

- [12] A. Moschitti, M. Nicosia, G. Barlacchi, Sacry: syntax-based automatic crossword puzzle resolution system, in: *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, 2015, pp. 79–84.
- [13] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, Crossword space: Latent manifold learning for italian crosswords and beyond, in: *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025)*, 2025, pp. 245–255.
- [14] G. Sarti, M. Nissim, It5: Text-to-text pretraining for italian language understanding and generation, in: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 9422–9433.
- [15] G. Sarti, M. Nissim, IT5: Text-to-text pretraining for Italian language understanding and generation, in: N. Calzolari, M.-Y. Kan, V. Hoste, A. Lenci, S. Sakti, N. Xue (Eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, ELRA and ICCL, Torino, Italy, 2024, pp. 9422–9433. URL: <https://aclanthology.org/2024.lrec-main.823>.
- [16] S. Kulshreshtha, O. Kovaleva, N. Shivagunde, A. Rumshisky, Down and across: Introducing crossword-solving as a new NLP benchmark, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2648–2659. URL: <https://aclanthology.org/2022.acl-long.189/>. doi:10.18653/v1/2022.acl-long.189.

A. Online Resources

The following resources are available:

- IT5 models for retrieving answers where partial solution is available
- IT5 models for retrieving answers
- Code

UniTor at Cruciverb-IT: Retrieval-Augmented Two-Step Reasoning for Italian Crossword Clue Answering

Andriy Shcherbakov¹, Danilo Croce² and Roberto Basili²

¹Reveal S.R.L., via Kenia 21, 00144, Rome, Italy

²Department of Enterprise Engineering, University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy

Abstract

Crossword clue answering requires the integration of lexical knowledge, semantic reasoning, and strict structural constraints. We present **UniTor**, an LLM-based system for Italian crossword clue answering, submitted to the CRUCIVERB-IT Task 1 at EVALITA 2026. UniTor formulates clue solving as a constrained ranking problem and combines retrieval-augmented evidence with a structured two-step prompting strategy executed within a single model call. In the first step, the model generates a diverse set of plausible candidate answers; in the second step, it enforces length constraints and re-ranks the candidates using explicit confidence scores. Extensive ablation experiments demonstrate that retrieval-based grounding and two-step reasoning provide complementary benefits, improving both recall and ranking stability across models of different sizes. Without any task-specific fine-tuning, UniTor achieves the highest score in the official evaluation.

Keywords

Crossword Solving, Large Language Models, Information Retrieval, Reasoning, Italian NLP

1. Introduction

Crossword clue answering poses a challenging problem for Natural Language Processing, as it requires mapping short, often indirect textual clues to correct solutions under strict structural constraints. Unlike standard question answering, crossword clues rarely correspond to explicit definitions and frequently exploit polysemy, figurative language, abbreviations, cultural references, and domain-specific conventions. Moreover, candidate solutions must satisfy hard constraints such as an exact character length and, in grid-based settings, cross-letter consistency.

The CRUCIVERB-IT shared task [1] at EVALITA 2026 [2] formalizes this problem as a ranking task: given an Italian clue c and a target answer length ℓ , systems are required to output a list of candidate solutions ordered by probability. This formulation reflects realistic crossword-solving conditions, where multiple candidates may be linguistically plausible and the correct solution is often determined only by additional constraints. For instance, for the clue “*Un quadro composto da frammenti*”¹, both *Collage* and *Mosaico* are valid semantic interpretations, and disambiguation cannot rely on meaning alone. As a result, crossword solving is more naturally modeled as a constrained ranking problem rather than as single-label prediction.

Previous work on Italian crosswords has shown that generic lexical or semantic similarity methods are insufficient to capture the complex relationship between clues and solutions. In particular, [3] frames clue answering as a specialized information retrieval problem, highlighting how standard matching approaches struggle when clues involve non-literal language or crossword-specific reasoning. These observations motivate approaches that combine high-recall retrieval with more structured inference and selection mechanisms.

Large Language Models (LLMs) constitute a promising foundation for this task, as they encode extensive lexical knowledge and exhibit strong associative and compositional reasoning abilities. Scaling-

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 - 27, Bari, IT

✉ shcherbakov@revealsrl.it (A. Shcherbakov); croce@info.uniroma2.it (D. Croce); basili@info.uniroma2.it (R. Basili)

ORCID 0000-0001-9111-1950 (D. Croce); 0000-0001-5140-0694 (R. Basili)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹Italian clue translation: “A picture composed of fragments.” The candidate answers are *Collage*, translated as “collage,” and *Mosaico*, translated as “mosaic.”

law studies suggest that increasing model size improves general language competence and knowledge coverage [4, 5]. However, when applied naively to constrained generation tasks, LLMs often exhibit undesirable behaviors, including violations of hard constraints, hallucination of rare or invalid forms, and unstable rankings across runs. These issues are particularly critical in CRUCIVERB-IT, where systems are evaluated on top- k accuracy and ranking quality (MRR) [1].

Two complementary directions have been proposed to mitigate these limitations. First, retrieval-augmented generation (RAG) grounds model outputs in external evidence, reducing reliance on parametric memory and improving robustness [6]. Second, structured prompting strategies, such as chain-of-thought and self-consistency, improve reasoning reliability by decomposing inference into multiple stages or separating generation from selection [7, 8]. Recent work further suggests that eliciting explicit probability distributions or diversified outputs can mitigate mode collapse and improve ranking consistency [9].

In this paper, we present **UniTor**, our submission to the CRUCIVERB-IT Task 1 [1]. UniTor is an LLM-based crossword clue solver built around three key design principles. First, it leverages retrieval-augmented suggestions obtained from a large repository of clue–solution pairs, using a length-constrained retrieval pipeline over a large clue–solution repository (we evaluated lexical, neural, and hybrid variants; the final configuration uses neural retrieval). Second, it adopts a two-step reasoning strategy that explicitly separates candidate exploration from constraint-aware re-ranking within a single LLM call. Third, it requires the model to output explicit probability scores for each candidate, aiming to stabilize rankings and support controlled diversity.

Beyond introducing the system, we use the CRUCIVERB-IT task as a controlled experimental setting to investigate four practical research questions: (i) whether requesting probabilistic outputs improves ranking quality over simple list generation; (ii) whether two-step prompting outperforms single-pass inference; (iii) whether and to what extent model scale affects performance; and (iv) how retrieval-based evidence interacts with model scale and prompting strategies. Our final system achieves the best performance in the official evaluation, ranking first among all submitted systems.

In the remainder of the paper, we review related work (Section 2), describe the UniTor system in detail (Section 3), present experimental results and ablation studies (Section 4), and conclude with a discussion of future directions.

2. Related Work

Early approaches to automatic crossword solving predominantly relied on lexical resources, handcrafted features, and traditional information retrieval techniques — such as BM25, edit distance, and surface-level similarity measures — to match clues with candidate answers. Systems in this category typically leveraged large clue–answer databases and heuristic ranking strategies, achieving reasonable performance on straightforward definitions but exhibiting limited robustness when confronted with indirect formulations, wordplay, or culturally grounded clues.

In the English setting, canonical examples of this paradigm include the Proverb system [10], which combined a large crossword database with expert modules and constraint satisfaction techniques to solve American-style crosswords. Similarly, WebCrow [11] explored the use of the web as a primary knowledge source, reducing dependence on handcrafted knowledge modules while still relying on heuristic retrieval and ranking. Closely related work on language games framed clue answering as a knowledge-based association problem and exploited large cultural repositories to identify candidates [12]. In the Italian crossword setting, Barlacchi et al. [13] and Moschitti et al. [14] study clue answering as a retrieval-and-ranking problem, showing that learning-to-rank and syntax-aware ranking strategies can improve ordering quality over pure retrieval baselines.

More recent work has shifted toward neural representations and learned retrieval models. In particular, Crossword Space [3] models clues and solutions as distinct “modalities” embedded in a shared latent space, training dual-encoder architectures with contrastive objectives on large Italian crossword datasets. These approaches outperform generic encoders and traditional retrieval baselines, highlighting

the importance of domain-specific representations for capturing polysemy, lateral associations, and crossword-specific conventions, and showing promising generalization to previously unseen words.

In parallel, Large Language Models (LLMs) have recently been investigated as solvers for crossword puzzles and related constrained language tasks [15]. Although LLMs demonstrate strong capabilities in paraphrasing and associative inference, their direct application to crossword solving remains challenging due to violations of hard structural constraints such as exact answer length. For instance, Saha et al. [15] report systematic difficulties in enforcing character-count constraints, especially for rare targets. The cryptic crossword setting is even more challenging, as it requires wordplay-driven inference that current LLMs still handle unreliably [16].

Several strategies have been proposed to mitigate these limitations. Retrieval-Augmented Generation (RAG) grounds model outputs in external evidence, improving robustness and selection quality [6]. Prompting strategies that elicit explicit reasoning or decompose inference into stages (e.g., chain-of-thought and related approaches) have been shown to enhance reliability in constrained tasks [7, 8]. More broadly, techniques that encourage diversified outputs and reduce mode collapse can further support stable ranking under ambiguity [9]. However, comparatively little work has systematically analyzed the interaction between retrieval-based grounding, prompt structure, and probabilistic ranking in the specific context of crossword clue answering for non-English languages such as Italian.

Finally, recent shared-task settings emphasize crossword clue answering as a constrained ranking problem. In particular, CRUCIVERB-IT at EVALITA 2026 evaluates systems that must return probability-ordered candidate lists under exact length constraints, using metrics such as Accuracy@k and MRR [1]. This formulation encourages approaches that combine high-recall retrieval with robust ranking and constraint handling.

Our work builds on these contributions by integrating retrieval-based suggestion generation, multi-step prompting, and probabilistic ranking within a unified LLM-based framework tailored to CRUCIVERB-IT. Unlike approaches that rely solely on learned retrieval models or unconstrained LLM generation, our system uses retrieval as evidence while delegating fine-grained discrimination among close alternatives to an LLM-based re-ranking step, enabling controlled ablations that quantify the impact of individual components.

3. System Description

In CRUCIVERB-IT Task 1, each input instance is defined as a pair (c, ℓ) , where c is an Italian crossword clue and $\ell \in \mathbb{N}$ denotes the exact character length of the target solution. Given (c, ℓ) , a system is required to produce an ordered list of candidate solutions $S = \langle s_1, \dots, s_K \rangle$ together with associated scores $P = \langle p_1, \dots, p_K \rangle$, such that $p_i \in [0, 1]$ and $p_i \geq p_{i+1} \forall i \in [1, K - 1]$ [1].

The scores are interpreted as confidence estimates rather than as a probability distribution: they are not required to sum to one and are used exclusively for ranking purposes. This formulation explicitly models crossword solving as a ranking problem, in which multiple candidates may be linguistically plausible and the correct solution is expected to appear among the highest-ranked entries.

Normalization and structural validity. To enforce crossword-specific constraints, we define a normalization operator $\mathcal{N}(\cdot)$ that maps a string to a canonical form. *A string is considered canonical if it is fully uppercased and contains only Latin uppercase characters in the range A–Z.* Accordingly, normalization consists of converting the input to uppercase and removing all characters outside this range, including accents, apostrophes, spaces, hyphens, and other non-alphabetic symbols.

A candidate solution s is said to be *structurally valid* with respect to a target length ℓ if:

$$\text{valid}(s, \ell) \iff |\mathcal{N}(s)| = \ell$$

When a pattern (or schema) is provided, structural validity additionally requires that all fixed letters in the schema match the corresponding positions in $\mathcal{N}(s)$. In the CRUCIVERB-IT Task 1 setting considered in

this work, no schema constraint is provided; therefore, we enforce only normalization and exact-length constraints.

Example. For the input $c = \text{“Capitale della Norvegia”}^2$ and $\ell = 4$, a correct system output may begin with:

$$\langle (\text{OSLO}, 0.92), (\text{MOSS}, 0.21), (\text{ALTA}, 0.15), \dots \rangle.$$

Candidates such as OSLO_ or BODØ are mapped by $\mathcal{N}(\cdot)$ to OSLO and BODO, respectively, and are therefore *structurally valid*. In contrast, ASKER is *not structurally valid* because $|\mathcal{N}(\text{ASKER})| = 5 \neq 4$.

3.1. Overview of UniTor

UniTor is a retrieval-grounded, LLM-based solver that maps an input pair (c, ℓ) to a probability-ranked list of candidate solutions. The system combines two tightly integrated components: (i) a *suggestion stage*, which retrieves length-compatible evidence from a large repository of clue–solution pairs, and (ii) a *reasoning-and-ranking stage*, in which an LLM generates, filters, and orders candidate solutions conditioned on the retrieved evidence.

The retrieved suggestions act as lightweight supervision in a few-shot-like manner, grounding the model’s generation process while preserving flexibility in candidate exploration.

Single-call, two-phase prompting. Although UniTor is conceptually described as a two-stage pipeline, the *reasoning-and-ranking stage is executed within a single LLM call*. Both candidate exploration and final selection are implemented through a single structured prompt that explicitly enforces a two-phase behavior, as detailed in Appendix A.

Concretely, the model is instructed to produce two ordered lists in sequence: (i) an exploratory list of *initial candidates*, designed to maximize recall and diversity and allowed to include candidates that do not satisfy the target length constraint, and (ii) a refined list of *final candidates*, obtained by filtering and re-ranking the initial candidates under hard structural constraints (exact length) and by assigning explicit probability scores.

This design ensures that exploratory reasoning and constraint-aware selection are both exposed and controlled, while avoiding multiple round-trips to the LLM.

Design objectives. The overall architecture of UniTor is driven by three complementary objectives: (a) *high recall* of plausible crossword solutions, achieved through retrieval-augmented suggestions; (b) *ranking quality and stability*, obtained by explicitly separating exploration from selection within the prompt; and (c) *constraint adherence*, enforced through normalization rules and optional post-processing.

3.2. Retrieval-Augmented Suggestions as Few-shot Evidence

UniTor relies on a large indexed repository $\mathcal{D} = \{(d_i, w_i, \ell_i, v_i)\}_{i=1}^N$ of approximately $N \approx 375,000$ word–definition pairs, where d_i denotes a definition or crossword clue, w_i the corresponding solution word, $\ell_i = |\mathcal{N}(w_i)|$ its normalized length, and v_i a dense vector representation computed using the BGE-M3 embedding model [17].

At inference time, given an input instance (c, ℓ) , the search space is first restricted to length-compatible entries:

$$\mathcal{D}_\ell = \{(d_i, w_i, \ell_i, v_i) \in \mathcal{D} : \ell_i = \ell\}.$$

From this subset, the system retrieves a small set of relevant examples $\mathcal{R}(c, \ell)$.

Crucially, retrieved items are not treated as final answers, but as *few-shot evidence* injected into the LLM prompt. Each retrieved pair provides lightweight in-context supervision of the form *definition* $\rightarrow$ *solution*, biasing the model toward crossword-consistent lexical choices while preserving generative flexibility.

² “Capital of Norway”

Retrieval strategies. We experiment with three retrieval strategies over $\mathcal{D}_\ell$:

- **Lexical (BM25):** candidates are ranked according to BM25 similarity between the input clue c and the stored definition d_i [18].
- **Neural:** the clue is encoded as $v_c = E(c)$, where $E(\cdot)$ denotes the BGE-M3 encoder, and candidates are ranked by cosine similarity $\cos(v_c, v_i)$.
- **Hybrid:** the top- M candidates retrieved with BM25 are re-ranked using a combined score, e.g., $\text{BM25}(c, d_i) \cdot \cos(v_c, v_i)$.

Example. For the clue $c = \text{“Un quadro composto da frammenti”}$ ³ and target length $\ell = 7$, the retrieval stage may return suggestions such as⁴:

$$\mathcal{R}(c, 7) = \{ (\text{MOSAICO}, \text{Composizione ottenuta assemblando tessere o frammenti}), \\ (\text{COLLAGE}, \text{Quadro realizzato incollando ritagli di carta o materiali diversi}), \dots \}$$

These examples are injected into the prompt as contextual evidence. The LLM is then asked to reason over them, disambiguate between semantically related candidates, and produce a probability-ranked output consistent with crossword conventions.

3.3. LLMs and Model Scale

We evaluate UniTor using a set of instruction-tuned Large Language Models (LLMs) that differ substantially in parameter scale, in order to assess whether performance gains in crossword clue answering are primarily driven by raw model capacity, retrieval grounding, or prompting and reasoning strategies.

This comparison is motivated by well-established scaling laws for neural language models, which suggest that larger models tend to encode richer lexical and world knowledge and exhibit stronger in-context learning capabilities. At the same time, crossword solving is a highly constrained task, in which structural validity and ranking stability may benefit more from explicit guidance than from sheer model size.

Concretely, we evaluate the following model families:

- **Llama 3.1 8B Instruct** [19], representing a relatively small but efficient open-weight model.
- **GPT-OSS 20B** and **GPT-OSS 120B** [20], allowing direct comparison between two scales within the same architectural and training paradigm.
- **GLM 4.6** [21], a very large instruction-tuned model with strong reported reasoning capabilities.

By evaluating the same retrieval-augmented, two-step prompting strategy across these models, we aim to disentangle the contribution of model scale from that of evidence injection and structured reasoning. In particular, this setup allows us to test whether smaller models can approach the performance of much larger ones when coupled with retrieval-based few-shot evidence and stabilized through explicit re-ranking.

3.4. Two-Step LLM Reasoning and Ranking

Rather than prompting the LLM to directly output a final ranked list, UniTor enforces a two-step reasoning process *within a single model request*. This design follows the general intuition of structured reasoning prompts and decomposition-based inference [7], while avoiding multiple interactions with the model.

The two steps are explicitly encoded in the prompt and reflected in the structured output, making the intermediate reasoning observable and controllable.

³ “A picture composed of fragments.”

⁴ (MOSAIC, composition made by assembling tiles or fragments), (COLLAGE, picture made by pasting paper cut-outs or different materials)

Step 1: Candidate proposal (exploration). In the first phase, the LLM receives the input pair (c, ℓ) together with the retrieved set $\mathcal{R}(c, \ell)$, which is injected as few-shot evidence. The model is instructed to generate up to K *plausible* candidate solutions, prioritizing semantic coverage and diversity rather than strict constraint satisfaction.

This exploratory step encourages the inclusion of: (i) synonyms and near-synonyms, (ii) hypernyms and related lexical items, and (iii) crossword-specific variants (e.g., preferred lemmas or crosswordese). Candidates proposed at this stage may violate the target length constraint and are assigned preliminary confidence scores.

Step 2: Re-ranking under constraints (selection). In the second phase, the LLM is instructed to re-evaluate the candidate set produced in Step 1 and to perform explicit selection and ranking. Concretely, the model is asked to: (i) apply normalization and enforce exact length constraints, (ii) compare candidates explicitly with respect to clue semantics and crossword conventions, and (iii) output a final list of candidates ordered by decreasing probability.

This separation between exploration and selection is intended to improve ranking stability and reduce the impact of spurious or overly salient candidates, in line with prior observations on robustness through selection and re-ranking in LLM reasoning.

3.5. Probabilistic Ranking, Constraints, and Post-processing

The final system output is a JSON-formatted list of pairs (s_i, p_i) , where s_i is a candidate solution and $p_i \in [0, 1]$ is an associated score, with candidates sorted in descending order of p_i . The scores are interpreted as *ranking confidences* rather than as a calibrated probability distribution, and therefore are not constrained to sum to one.

Explicit probability assignment serves two purposes: it aligns with the task specification, and it enables more stable ranking behavior by encouraging the model to distribute confidence across multiple plausible solutions rather than collapsing onto a single mode, as suggested by previous work on self-consistency and diversity in LLM outputs [9].

Constraint handling. In UniTor, we consider two constraint-handling configurations:

- **Intrinsic (LLM-only) constraint adherence:** constraints on normalization and length are enforced solely through the prompt, without external filtering. This setting allows us to directly measure the model’s ability to internalize and respect hard structural constraints.
- **Post-processed constraint enforcement:** the normalization operator $\mathcal{N}$ is applied programmatically, and any candidate violating $\text{valid}(s, \ell)$ is discarded. If fewer than K candidates remain, additional length-valid candidates may be backfilled from the retrieval set $\mathcal{R}(c, \ell)$.

Post-processing can improve the task scores, and it partially shifts the burden from generation to retrieval. In Section 4 we report results for the first configuration in order to disentangle intrinsic reasoning capabilities from external constraint enforcement.

Summary. UniTor formulates crossword clue answering as a *constrained ranking problem* rather than as single-label prediction. The system combines (i) length-filtered retrieval to provide retrieval-augmented few-shot evidence, (ii) a structured two-step prompting strategy implemented within a single LLM call to separate exploration from selection and improve ranking stability, and (iii) explicit probabilistic outputs to support robust ordering of multiple plausible candidates. This design directly matches the requirements of the CRUCIVERB-IT task [1] and allows us to systematically analyze the impact of retrieval grounding, reasoning decomposition, and probabilistic ranking on crossword clue answering performance. Appendix A provides the full UniTor prompt and the exact output format used in our experiments.

4. Experimental Results

The effectiveness of UniTor has been evaluated through a series of controlled ablation tests whose results are derived from the official CRUCIVERB-IT Task 1 test set. Our experiments are designed to quantify the individual and combined impact of four key design choices: (i) explicit probabilistic outputs, (ii) two-step reasoning within a single LLM call, (iii) retrieval-augmented suggestions, and (iv) model scale.

Experimental setting. UniTor does not rely on any form of task-specific fine-tuning. All experiments are conducted using off-the-shelf, instruction-tuned large language models accessed via prompt-based inference (through the `ollama` framework⁵), without updating model parameters or performing supervised adaptation. This design choice allows us to focus the analysis on the contribution of individual system components—prompt structure, retrieval grounding, and ranking strategy—rather than on training effects.

For ablation experiments, we use the official CRUCIVERB-IT dataset⁶ released for EVALITA 2026. Specifically, we randomly sample 1,000 instances from the test split and use them as a validation set for controlled comparisons across different prompt configurations and model choices.

The full training portion of the dataset (approximately 375,000 word–definition pairs) is not used for any model training. Instead, it is employed during a preparatory phase to build the retrieval index used for suggestion generation. The index is implemented using the Qdrant vector database⁷ and stores, for each entry, the clue text (definition), the corresponding solution (word) along with its character length. Each clue is further enhanced with a dense representation computed using the BGE-M3 embedding model [17], enabling neural and hybrid retrieval.

Retrieval variants and selection criterion. We experimented with three retrieval strategies: (i) a purely lexical retriever based on BM25, (ii) a neural retriever based on dense embeddings (cosine similarity in the BGE-M3 space), and (iii) a hybrid strategy that combines the two signals by multiplying their scores.⁸ Importantly, retrieval quality was not evaluated as a standalone IR task. Instead, we compared retrieval strategies within the full end-to-end pipeline, measuring their impact on the final crossword-solving metrics (Accuracy@1, Accuracy@10 and MRR).

In these experiments, the neural retriever yielded the strongest downstream performance. For space reasons, we report only the neural-retrieval configuration in the main paper, while the BM25 and hybrid variants are omitted.

For prompt configurations requiring 10, 20, or 50 suggestions, we retrieve candidates by enforcing the answer length as a hard constraint and inject the resulting clue–solution examples as contextual evidence. For each prompt configuration, this procedure produces 1,000 instantiated prompts, which are then submitted to the corresponding LLM for inference.

Output validation and formatting issues. After inference, we parse model outputs to extract the candidate lists and discard any extraneous text outside the prescribed format. When a specific output format is required, a small fraction of generations may be malformed (e.g., due to missing brackets, quotes, or comma separators, or because JSON-like content is mixed with comments). In these cases, we apply minimal, deterministic syntactic repairs to recover a valid serialization (e.g., balancing brackets or quotes, or removing inline comments), without altering the set or order of predicted candidate strings.

The frequency of malformed outputs decreases with increasing model size: GLM 4.6 rarely required intervention; GPT-OSS 20B produced approximately 15–20 invalid outputs out of 1,000 (1.5–2%); Llama 3.1 8B produced approximately 25% invalid outputs and was therefore excluded from the detailed analysis.

⁵<https://ollama.com/>

⁶<https://huggingface.co/datasets/cruciverb-it/evalita2026>

⁷<https://qdrant.tech/>

⁸In all cases, answer length is enforced as a hard constraint at retrieval time.

Results. The CRUCIVERB-IT Task 1 evaluates systems in a *constrained ranking* setting: given a clue and a target answer length, systems must output an ordered list of candidates. Performance is measured using ranking-aware metrics, namely Accuracy@ k (whether the gold solution appears within the top- k ranked candidates, with $k \in \{1, 10\}$) and MRR (Mean Reciprocal Rank), which accounts for the position of the first correct candidate in the ranked list (higher is better) [1].

Table 1 reports an ablation study conducted without retrieval-augmented suggestions, isolating the effect of prompt design choices and model scale. In particular, we compare: (i) the use of explicit probabilistic outputs versus greedy list generation, and (ii) a single-step versus a two-step reasoning strategy, across three LLMs of increasing size.

Prob.	Two-step	Accuracy@1			Accuracy@10			MRR		
		GPT-OSS 20B	GPT-OSS 120B	GLM 4.6	GPT-OSS 20B	GPT-OSS 120B	GLM 4.6	GPT-OSS 20B	GPT-OSS 120B	GLM 4.6
No	No	26.60	45.10	44.60	35.60	55.30	61.50	30.02	49.28	51.09
	Yes	25.80	44.30	50.00	29.50	49.80	60.70	27.36	46.81	54.56
Yes	No	24.80	45.50	48.00	32.30	55.30	62.40	27.85	49.67	54.16
	Yes	23.40	44.10	53.80	27.40	50.00	64.40	24.98	46.64	58.29

Table 1

Performance (%) of UniTor variants *without retrieval-augmented suggestions*, analyzing the effect of probabilistic outputs and two-step prompting across different LLMs.

Several trends emerge from the results. First, model scale has a strong and consistent impact across all configurations: larger models substantially outperform smaller ones, confirming that crossword clue answering benefits from increased parametric knowledge and reasoning capacity. However, scale alone is not sufficient to guarantee stable or optimal ranking quality.

Second, the introduction of a two-step reasoning strategy yields clear improvements in ranking-oriented metrics, particularly MRR, for the largest model (GLM 4.6). This suggests that explicitly separating candidate generation from selection helps mitigate ranking instability and overconfident early choices, even in the absence of external evidence.

Third, explicit probabilistic outputs have a nuanced effect. While they do not consistently improve Accuracy@1 in isolation, they tend to benefit ranking quality (MRR and Accuracy@10), especially when combined with two-step prompting. This supports the hypothesis that requiring the model to distribute confidence across multiple candidates reduces mode collapse and improves relative ordering among plausible solutions.

Finally, interactions between prompt structure and model scale are non-trivial. For smaller and mid-sized models, additional structure can sometimes slightly reduce Accuracy@1, likely due to increased cognitive load. In contrast, larger models are able to exploit structured prompts more effectively, translating them into measurable gains in ranking quality.

Table 2 reports results obtained by augmenting the LLM with retrieval-based suggestions, varying both the number of retrieved examples and the use of single-step versus two-step prompting. Following the trends observed in Table 1, we always use the probabilistic output format in this set of experiments, as it provided the most consistent gains in ranking quality—especially for the largest model—while keeping the comparison focused on the effect of retrieval and prompt structure.

In this setting, retrieved clue–solution pairs are injected into the prompt as few-shot evidence, allowing us to assess how external grounding interacts with reasoning structure and model scale.

The results show that retrieval-augmented suggestions have a large and consistent impact on performance across all models and metrics. Even a small number of retrieved examples (10) leads to substantial gains over the no-retrieval baseline, with Accuracy@1 improvements exceeding 10 percentage points for all models.

Increasing the number of retrieved suggestions generally improves performance, but with diminishing returns. Gains between 10 and 20 examples are modest, and in some cases performance plateaus or

Strategy		Accuracy@1			Accuracy@10			MRR		
Two-step	Added Ex.	GPT-OSS 20B	GPT-OSS 120B	GLM 4.6	GPT-OSS 20B	GPT-OSS 120B	GLM 4.6	GPT-OSS 20B	GPT-OSS 120B	GLM 4.6
No	0	24.80	45.50	48.00	32.30	55.30	62.40	27.85	49.67	54.16
	10	51.50	62.80	63.30	64.50	76.10	76.30	56.65	68.15	68.57
	20	51.60	62.00	63.80	65.80	77.60	74.50	57.00	68.22	68.24
	50	50.80	65.60	66.20	63.90	79.90	76.80	55.95	71.26	70.52
Yes	0	24.40	43.80	53.80	28.60	51.30	63.70	26.14	47.03	58.01
	10	47.10	59.40	65.50	58.10	71.40	75.20	51.60	64.46	69.68
	20	48.30	59.40	65.60	57.60	70.50	76.20	52.09	64.33	69.92
	50	49.10	62.10	66.80	59.20	71.80	77.50	53.06	66.15	71.31

Table 2

Performance (%) of retrieval-augmented suggestions as a function of the number of injected examples (Added Ex.) and the prompting strategy (two-step vs. single-step), evaluated across three instruction-tuned LLMs.

slightly degrades when moving to 50 examples.

Crucially, the interaction between retrieval and two-step prompting is systematic. While retrieval alone significantly boosts recall-oriented metrics (Accuracy@10), the two-step strategy consistently improves ranking quality, as reflected in higher MRR values, especially for larger models.

Model scale remains an important factor, but its relative contribution changes in the presence of retrieval. With sufficient external evidence, smaller models close a large portion of the gap with larger ones, suggesting that retrieval-based few-shot supervision partially compensates for limited parametric capacity.

Overall, these results support the view that retrieval and structured reasoning play complementary roles: retrieval improves coverage and recall by constraining the hypothesis space, while two-step prompting acts as an implicit re-ranking and verification mechanism, improving stability and ranking quality.

Final configuration and official results. Table 2 shows that increasing the number of retrieval-augmented suggestions from $k = 10$ to $k = 50$ yields only modest performance gains, while substantially increasing prompt length and, consequently, inference cost and context pressure. For this reason, in the final submission we set $k = 10$, keeping the retrieval context compact while retaining most of the benefits of few-shot evidence injection. The submitted system therefore combines (i) retrieval-augmented few-shot evidence with $k = 10$, (ii) a single-call two-step prompting strategy, and (iii) explicit probabilistic ranking.

Table 3 reports the official results for CRUCIVERB-IT Task 1. UniTor achieves an Accuracy@1 of 0.69, an Accuracy@10 of 0.83, and an MRR of 0.72, substantially outperforming both the shared-task baseline and the average score across submitted runs.

The baseline provided by the task organizers frames clue answering as a pure information retrieval problem: for each test clue, it retrieves the top-10 most similar training clues using BM25 and returns the corresponding answers as candidates. While this approach performs well for near-duplicate clues, it struggles when the clue-solution relationship relies on paraphrasing, lexical substitution, polysemy, or crossword-specific conventions.

UniTor improves substantially over this baseline by combining retrieval with explicit LLM-based selection. Retrieved items act as length-compatible evidence and lightweight supervision, while the two-step prompting strategy enforces a clear separation between candidate exploration and constraint-aware re-ranking. This combination yields a marked improvement both in top-1 accuracy (+0.29

Table 3

Top-performing systems in the official CRUCIVERB-IT Task 1 evaluation.

System	Accuracy@1	Accuracy@10	MRR
UniTor	0.69	0.83	0.72
Second-best system	0.58	0.75	0.63
Average score (all runs)	0.52	0.70	0.57
Baseline	0.40	0.62	0.46

absolute over the baseline) and in ranking quality (MRR +0.26), indicating that the gains stem not only from improved recall but also from more reliable ordering of plausible candidates.

Beyond outperforming the baseline, UniTor shows a clear margin over the second-best submitted system across all evaluation metrics (Table 3). In particular, UniTor improves Accuracy@1 by +0.11 absolute and MRR by +0.09, demonstrating that the advantage is not limited to higher recall at larger cutoffs, but reflects consistently better placement of the correct solution among the top-ranked candidates.

This performance gap is noteworthy given that all systems operate under the same constraints, namely the absence of grid information and the availability of only the target answer length. The results suggest that UniTor’s advantage does not arise solely from stronger retrieval, but from the interaction between retrieval and LLM-based selection: the two-step prompting strategy explicitly separates candidate generation from constraint-aware re-ranking, leading to more stable and discriminative rankings.

Compared to the average score across all submitted runs, UniTor achieves gains of +0.17 in Accuracy@1 and +0.15 in MRR. This consistent margin indicates that the proposed architecture generalizes beyond a single favorable configuration and that its components contribute synergistically rather than in isolation. Our ablation studies further suggest that the strongest results are obtained when the two-step probabilistic prompting strategy is paired with a highly capable LLM (in our case, GLM). While GPT-OSS-based configurations already outperform the baseline and most competing systems, the stronger reasoning and selection capabilities of GLM appear to amplify the benefits of retrieval-augmented evidence and explicit re-ranking. We hypothesize that a GPT-OSS-based variant using the same final configuration ($k = 10$, two-step reasoning, probabilistic output) would still yield competitive results, albeit with a smaller margin over the second-best system, consistent with the trends observed in our validation experiments.

5. Conclusions

In this paper we presented UniTor, an LLM-based system for Italian crossword clue answering submitted to the CRUCIVERB-IT Task 1 at EVALITA 2026. UniTor approaches clue solving as a constrained ranking problem and combines three complementary ingredients: retrieval-augmented few-shot evidence, a structured two-step reasoning strategy implemented within a single LLM call, and explicit probabilistic ranking of candidate solutions. Importantly, the system operates without any task-specific fine-tuning, relying exclusively on prompt design, retrieval grounding, and model-internal reasoning.

Our experimental analysis demonstrates that each component plays a distinct and synergistic role. Retrieval-augmented suggestions provide a strong inductive bias toward crossword-consistent lexical choices and yield large gains in recall-oriented metrics even with a small number of retrieved examples. Two-step prompting improves ranking stability and discrimination among highly plausible candidates, particularly for larger models, by explicitly separating candidate exploration from constraint-aware selection. Explicit probability assignment further enhances ranking quality, especially in combination with structured reasoning, supporting the view that distributional outputs mitigate mode collapse and improve relative ordering in ambiguous settings.

On the official CRUCIVERB-IT Task 1 evaluation, UniTor achieves state-of-the-art performance, substantially outperforming both the BM25-based baseline and all competing systems across Accuracy@1, Accuracy@10, and MRR. The large margin over the baseline confirms that pure retrieval is insufficient to solve crossword clues reliably, while the consistent gap over the second-best system indicates that the gains are not merely due to improved recall, but to more effective ranking and selection. Ablation results further show that retrieval-based grounding allows smaller models to close a significant portion of the performance gap with larger ones, while the strongest results are obtained when retrieval, two-step reasoning, and probabilistic ranking are paired with a highly capable LLM.

This work focuses exclusively on clue answering in isolation and does not address grid-level crossword solving. Integrating UniTor into a full crossword solver that exploits crossing constraints and global grid consistency represents a natural next step, potentially allowing the ranking scores produced by UniTor to act as soft constraints in a global optimization framework. More broadly, our results suggest that for knowledge-intensive, weakly specified NLP tasks, carefully designed prompting and retrieval strategies can rival or surpass more complex training-based approaches. We hope that UniTor provides a useful reference point for future work on crossword solving and on structured, retrieval-grounded reasoning with large language models.

Acknowledgments

The authors acknowledge financial support from the PNRR MUR project PE0000013-FAIR. Moreover, the authors acknowledge support from Project ECS 0000024 Rome Technopole, - CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, Funded by the European Union - NextGenerationEU.

Declaration on Generative AI

Parts of the writing and editing process for this manuscript were supported by the use of generative AI tools, specifically OpenAI’s ChatGPT. These tools were employed to enhance clarity and correct spelling. All AI-assisted content was carefully reviewed and validated by the authors to ensure accuracy, originality, and compliance with ethical and scientific standards. The authors bear full responsibility for the final content.

References

- [1] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, M. Nissim, Cruciverb-it @ evalita 2026: Overview of the crossword solving in italian task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Apro시오, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] C. Ciaccio, G. Sarti, A. Miaschi, F. Dell’Orletta, Crossword space: Latent manifold learning for Italian crosswords and beyond, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, p. 245–255. URL: <https://aclanthology.org/2025.clicit-1.26/>.
- [4] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, 2020. URL: <https://arxiv.org/abs/2001.08361>. arXiv: 2001.08361.
- [5] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, T. Hennigan, E. Noland, K. Millican, G. van den Driessche, B. Damoc,

- A. Guy, S. Osindero, K. Simonyan, E. Elsen, J. W. Rae, O. Vinyals, L. Sifre, Training compute-optimal large language models, 2022. URL: <https://arxiv.org/abs/2203.15556>. arXiv:2203.15556.
- [6] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL: <https://arxiv.org/abs/2005.11401>. arXiv:2005.11401.
- [7] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, 2023. URL: <https://arxiv.org/abs/2201.11903>. arXiv:2201.11903.
- [8] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, 2023. URL: <https://arxiv.org/abs/2203.11171>. arXiv:2203.11171.
- [9] J. Zhang, S. Yu, D. Chong, A. Sicilia, M. R. Tomz, C. D. Manning, W. Shi, Verbalized sampling: How to mitigate mode collapse and unlock llm diversity, 2025. URL: <https://arxiv.org/abs/2510.01171>. arXiv:2510.01171.
- [10] M. L. Littman, G. A. Keim, N. Shazeer, A probabilistic approach to solving crossword puzzles, *Artificial Intelligence* 134 (2002) 23–55. URL: <https://www.sciencedirect.com/science/article/pii/S000437020100114X>. doi:[https://doi.org/10.1016/S0004-3702\(01\)00114-X](https://doi.org/10.1016/S0004-3702(01)00114-X).
- [11] G. Angelini, M. Ernandes, M. Gori, Webcrow: A web-based crosswords solver, in: M. Maybury, O. Stock, W. Wahlster (Eds.), *Intelligent Technologies for Interactive Entertainment*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 295–298.
- [12] P. Basile, M. de Gemmis, P. Lops, G. Semeraro, Solving a complex language game by using knowledge-based word associations discovery, *IEEE Transactions on Computational Intelligence and AI in Games* 8 (2014) 1–1. doi:10.1109/TCIAIG.2014.2355859.
- [13] G. Barlacchi, M. Nicosia, A. Moschitti, Learning to rank answer candidates for automatic resolution of crossword puzzles, in: R. Morante, S. W.-t. Yih (Eds.), *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, Ann Arbor, Michigan, 2014, pp. 39–48. URL: <https://aclanthology.org/W14-1605/>. doi:10.3115/v1/W14-1605.
- [14] A. Moschitti, M. Nicosia, G. Barlacchi, SACRY: Syntax-based automatic crossword puzzle resolution sYstem, in: H.-H. Chen, K. Markert (Eds.), *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, Association for Computational Linguistics and The Asian Federation of Natural Language Processing, Beijing, China, 2015, pp. 79–84. URL: <https://aclanthology.org/P15-4014/>. doi:10.3115/v1/P15-4014.
- [15] S. Saha, S. Chakraborty, S. Saha, U. Garain, Language models are crossword solvers, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 2074–2090. URL: <https://aclanthology.org/2025.naacl-long.104/>. doi:10.18653/v1/2025.naacl-long.104.
- [16] A. Sadallah, D. Kotova, E. Kochmar, What makes cryptic crosswords challenging for LLMs?, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 5102–5114. URL: <https://aclanthology.org/2025.coling-main.342/>.
- [17] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, Z. Liu, M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2025. URL: <https://arxiv.org/abs/2402.03216>. arXiv:2402.03216.
- [18] S. Robertson, H. Zaragoza, The probabilistic relevance framework: Bm25 and beyond, *Found. Trends Inf. Retr.* 3 (2009) 333–389. URL: <https://dl.acm.org/doi/abs/10.1561/15000000019>.
- [19] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, al., The llama 3 herd of models, 2024. URL: <https://arxiv.org/abs/2407.21783>. arXiv:2407.21783.
- [20] OpenAI, S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, al., gpt-oss-120b & gpt-oss-20b

model card, 2025. URL: <https://arxiv.org/abs/2508.10925>. arXiv:2508.10925.

- [21] GLM-4.5 Team, A. Zeng, X. Lv, Q. Zheng, Z. Hou, B. Chen, C. Xie, C. Wang, D. Yin, H. Zeng, al., Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL: <https://arxiv.org/abs/2508.06471>. arXiv:2508.06471.

A. Appendix

This appendix reports the prompt templates used to query the Large Language Models in our experiments. The templates are systematically constructed by varying three orthogonal dimensions: (i) the prompt structure, distinguishing between a single-step (plain) formulation and a two-step formulation; (ii) the inclusion or exclusion of explicit scoring, where the model is instructed to assign confidence or probability values to candidate solutions; and (iii) the maximum number of requested suggestions, set to 0, 10, 20, or 50. Together, these dimensions define the set of prompt configurations evaluated in the study. This template corresponds to a single-step prompt setting in which no suggestions are provided and no explicit scoring of candidate answers is requested:

```
Sei un esperto risolutore di cruciverba in italiano.  
Riceverai in input una definizione. Il tuo compito e' fornire fino a <N> possibili soluzioni, ognuna  
  composta esattamente da <L> caratteri.  
Scrivi ogni proposta su una nuova riga, senza numerarle e senza aggiungere commenti.  
Definizione: <D>
```

This template corresponds to a single-step prompt setting without retrieval-based suggestions, in which the scoring of candidate answers is explicitly requested.

```
Sei un esperto di cruciverba in italiano.  
Riceverai una definizione e dovrai fornire fino a <N> possibili soluzioni, indicando per ciascuna una  
  probabilità compresa tra 0 e 1 (numero decimale) che sia quella corretta.  
  
Ogni soluzione deve:  
- essere composta esattamente da <L> caratteri  
- essere riportata su una nuova riga nel formato:  
  <soluzione>, <probabilità>  
  
Non aggiungere numerazioni, commenti o spiegazioni. Ordina tutte le proposte in ordine decrescente di  
  probabilità.  
Definizione: <D>
```

This template represents a single-step prompt with up to R retrieval-based suggestions and explicit candidate scoring.

```
Sei un esperto di cruciverba in italiano.  
Riceverai:  
- una definizione principale, per la quale dovrai proporre fino a <N> soluzioni possibili, indicando  
  per ciascuna una probabilità compresa tra 0 e 1 (numero decimale) che sia corretta;  
- fino a <R> coppie (parola - definizione), da considerare come informazioni aggiuntive potenzialmente  
  utili.  
  
Ogni soluzione deve:  
- essere composta esattamente da <L> caratteri  
- essere riportata su una nuova riga nel formato:  
  <soluzione>, <probabilità>  
  
Non aggiungere numerazioni, commenti o spiegazioni.  
Ordina le proposte in ordine decrescente di probabilità.  
Definizione: <D>
```

This template corresponds to a two-step prompt configuration with retrieval-based suggestions (with a maximum number of S), in which the explicit scoring of candidate answers is required.

Each prompt enforces a two-phase generation process: (i) an exploratory stage (`candidati_iniziali`), in which up to N (set to 10) semantically plausible candidates are proposed without enforcing length constraints; and (ii) a selection stage (`definitivi`), which applies normalization, exact length filtering, and optional schema constraints, followed by re-ranking of the remaining candidates. Retrieved clue–solution pairs (up to R) are injected as lightweight evidence to support candidate generation and selection.

```
Sei un esperto risolutore di cruciverba in lingua italiana.
Riceverai:
- una definizione testuale;
- un numero di caratteri (lunghezza esatta, solo lettere A-Z);
- fino a <R> definizioni (parola - definizione), che potrebbero essere utili.

## STRATEGIA A DUE FASI (con output doppio)
Genera due liste in sequenza:

1) FASE 1 - CANDIDATI_INIZIALI (brainstorming)
- Genera fino a <N> parole plausibili semanticamente rispetto alla definizione (sinonimi, iperonimi,
  tecnicismi, locuzioni normalizzate).
- Possono includere lunghezze errate rispetto a '[[LUNGHEZZA]]'. Non scartarle qui: questa è una
  fase esplorativa.
- Normalizza comunque tutte le voci (vedi regole).
- Calcola 'probabilita' secondo i pesi indicati (sotto).
- Ordina per probabilità decrescente e rimuovi duplicati.

2) FASE 2 - DEFINITIVI (filtro & affinamento)
- Prendi solo i candidati della FASE 1 che:
  - hanno 'LEN(normalizzata) == [[LUNGHEZZA]]';
- Ispirati ai candidati della FASE 1 e alla fonti per aggiungere:
  - sinonimi piu' precisi/usuali in ambito cruciverbistico,
  - varianti morfologiche corrette,
  - lemmi preferiti (singolare/maschile-forma base, salvo diversa indicazione).
- Escludi qualsiasi voce che non rispetti lunghezza e schema.
- Ricalcola le 'probabilita' con gli stessi pesi e ordina.
- Se nessuna voce è valida, restituisci "definitivi": [].

## NORMALIZZAZIONE (sempre, in entrambe le fasi)
- Tutto in MAIUSCOLO.
- Rimuovi accenti, apostrofi, spazi e trattini.
- Ammesse solo lettere A-Z.
- Il campo 'lunghezza' è la lunghezza della parola normalizzata.

## SCORING (per entrambe le fasi)
Assegna 'probabilita' in (0,1] come somma ponderata:
- 0.40 -> Schema/Lunghezza
  - Fase 1: +0.40 solo se lunghezza corretta e schema rispettato (se presente); altrimenti 0 su questa
    quota.
  - Fase 2: requisito obbligatorio (altrimenti la voce non entra nei 'definitivi').
- 0.35 -> Attinenza semantica alla definizione.
- 0.25 -> Plausibilità cruciverbistica (forma più comune, lemma preferito, uso frequente).

## CHECKLIST (prima di emettere l'output)
- [ ] Tutte MAIUSCOLE e normalizzate
- [ ] 'lunghezza' corrisponde alla parola normalizzata
- [ ] In FASE 2: lunghezza corretta e schema rispettato
- [ ] Nessun duplicato
- [ ] Nessun carattere non ammesso
- [ ] 'probabilita' in (0,1]

## DIVIETI
- Non aggiungere testo esplicativo fuori dallo JSON richiesto.
- Non includere parole troncate o con caratteri non ammessi.
```


- Se la definizione ha “?” -> trattala come definizione figurata/gioco di parole.
- Se è indicato “abbr.” o “sigla” -> proponi solo abbreviazioni/sigle.

IMPORTANTE - CASI DA 2 LETTERE (Regole d’Oro)

Se ‘[[LUNGHEZZA]] == 2’, non inventare nulla di semantico o fantasioso.

Quasi sempre si tratta di giochi di lettere o sigle.

Segui rigorosamente queste 10 REGOLE D’ORO:

1. "Inizio / Prime / Principio di ..." -> usa le prime due lettere.
2. "Fine / Ultime / Termine di ..." -> usa le ultime due lettere.
3. "Limiti / Confini / Estreme di ..." -> prendi prima + ultima lettera.
4. "In mezzo a / Nel centro di ..." -> prendi le due centrali (o la coppia interna evidente).
5. "Gemelle / Uguali / Doppie ..." -> prendi la lettera raddoppiata presente (es. “NN”, “LL”).
6. "Consonanti / Vocali di ..." -> conserva solo consonanti o solo vocali del termine.
7. "Sigla / Targa / Provincia / Simbolo" -> scegli abbreviazioni ufficiali o codici (AR, LI, PD, BO, AI, UV, KO, OK...).
8. "Iniziali di ..." / "(iniz.)" -> usa le prime lettere dei nomi indicati.
9. "Senza / Fra / Tra ..." -> togli o confronta lettere esplicitamente tra le parole.
10. Definizione secca (tipo “Preposizione”, “Negazione”, “Risposta affermativa”) -> usa lemmi di 2 lettere noti:
IN, NO, SI, SU, DA, DI, RE, RA, TO, OR, ME, IO, ecc.

FORMATO DI OUTPUT

Un unico oggetto JSON con due array in questo ordine:

- ‘candidati_iniziali’ -> fino a <N> candidati (possono includere lunghezze errate).
- ‘definitivi’ -> fino a <N> voci filtrate e migliorate (tutte con lunghezza corretta e schema rispettato).

Ogni elemento di entrambi gli array deve rispettare questo schema:

```
{ "parola": "^[A-Z]+$", "lunghezza": <int>, "probabilita": <decimal number from 0.0 to 1.0> }
```

Response Formats

crossword_double

// Oggetto con le due liste: brainstorming (fase 1) e soluzione finale (fase 2).

```
{
  "type": "object",
  "properties": {
    "candidati_iniziali": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "parola": { "type": "string", "pattern": "^[A-Z]+$" },
          "lunghezza": { "type": "integer", "minimum": 1 },
          "probabilita": { "type": "number", "minimum": 0, "maximum": 1 }
        },
        "required": ["parola", "lunghezza", "probabilita"],
        "additionalProperties": false
      }
    },
    "definitivi": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "parola": { "type": "string", "pattern": "^[A-Z]+$" },
          "lunghezza": { "type": "integer", "minimum": 1 },
          "probabilita": { "type": "number", "minimum": 0, "maximum": 1 }
        },
        "required": ["parola", "lunghezza", "probabilita"],
        "additionalProperties": false
      }
    }
  },
  "required": ["candidati_iniziali", "definitivi"],
  "additionalProperties": false
}

Definizione: <D>
Lunghezza: <L>
Informazione aggiuntiva: <up to R word/definition pairs>
```

SVELA at EVALITA 2026: Overview of the Selective Verification of Erasure from LLM Answers Task

Claudio Savelli^{1,*}, Moreno La Quatra², Alkis Koudounas¹ and Flavio Giobergia¹

¹Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italy

²Università degli Studi di Enna "Kore", Piazza dell'Università, 94100 Enna, Italy

Abstract

This paper presents SVELA (Selective Verification of Erasure from LLM Answers), a shared task at EVALITA 2026. SVELA challenges participants to develop methods that verify whether a Large Language Model has successfully forgotten specific information. Given models that have undergone unlearning, participants must classify fictional identities or individual facts as retained, forgotten, or never seen during training. The task provides two complementary subtasks: entity-level detection, where entire identities are classified, and instance-level detection, where individual question-answer pairs are evaluated. The task attracted eight registered teams, four of which submitted system description papers, and resulted in more than fifty valid submissions across the two subtasks. The evaluation highlights the intrinsic difficulty of unlearning verification, particularly at the instance level, where less aggregated information and more fine-grained distinctions between retain, forget, and never-seen information are required.

Keywords

Machine Unlearning, Evaluation Metrics, Large Language Models, Italian NLP, EVALITA 2026

1. Motivation

Large Language Models (LLMs) acquire vast amounts of information during training. This information may include personal data, copyrighted content, or other sensitive material that users or regulations may require to be removed [1]. Privacy regulations, such as the European Union's General Data Protection Regulation (GDPR), grant individuals the right to request the deletion of their personal data from automated systems [2]. However, retraining these models from scratch after removing specific data is prohibitively expensive, often requiring weeks of computation and millions of dollars [3].

Machine Unlearning (MU) offers a practical alternative by removing the influence of specific data from trained models without full retraining [4]. Several unlearning methods have been proposed for LLMs, including gradient-based approaches, optimization techniques, and preference-based alignment [5, 6]. These methods aim to make a model behave as if it had never seen the targeted data.

MU is commonly evaluated along three complementary dimensions: utility, measuring the extent to which model performance on retained data is preserved; efficacy, assessing whether the influence of the targeted data has been effectively removed; and efficiency, capturing the computational cost of the unlearning procedure. Recent benchmarking efforts have formalized this perspective by proposing unified evaluation frameworks that explicitly account for all three axes [7, 8, 9].

In this work, SVELA focuses specifically on the problem of verifying the efficacy of unlearning in LLMs. The task targets the downstream verification setting, where a trained model is given and the goal is to determine, through post-hoc analysis, whether specific information has been successfully forgotten. To this end, SVELA provides a controlled experimental setting for the development and comparison of unlearning verification methods.

The task relies on synthetic data describing fictional identities, ensuring that no real personal information is involved and that the ground truth is perfectly known. By covering multiple languages,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26-27, Bari, IT

*Corresponding author.

✉ claudio.savelli@polito.it (C. Savelli); moreno.laquatra@unikore.it (M. La Quatra); alkis.koudounas@polito.it (A. Koudounas); flavio.giobergia@polito.it (F. Giobergia)

id 0000-0002-0877-7063 (C. Savelli); 0000-0001-8838-064X (M. La Quatra)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

including Italian, Spanish, French, and German, SVELA enables the evaluation of verification methods in multilingual contexts. Participants are invited to design robust verification approaches capable of reliably distinguishing between information that a model retains, has forgotten, or has never been exposed to.

SVELA is proposed within the framework of EVALITA 2026 [10], the evaluation campaign for Natural Language Processing and Speech tools for Italian. This positioning underscores the importance of developing privacy-preserving AI technologies specifically tailored to the needs of the Italian NLP community. As European institutions, and Italian ones in particular, navigate the complexities of GDPR compliance, the ability to selectively forget data becomes a critical capability. By benchmarking unlearning methods on Italian data alongside those from other languages, SVELA supports EVALITA’s mission to advance the robustness and reliability of Italian-language technologies in an era of increasing regulatory scrutiny.

2. Task Definition

SVELA evaluates the ability of participant-developed methods to verify whether MU has succeeded. Unlike traditional unlearning tasks, where participants apply forgetting techniques, SVELA focuses on the complementary problem of detecting what a model knows or has forgotten.

2.1. Task Setup

Participants receive two components. First, they receive access to a set of LLMs $\mathcal{M} = \{M_1, M_2, \dots, M_k\}$ of different sizes. Each model has been fine-tuned on a set of fictional actor biographies and subsequently processed with a state-of-the-art unlearning method. The specific unlearning method applied to each model is hidden from participants. Second, participants receive a set of fictional identities, $\mathcal{I}$, along with questions that can be asked about them. Each identity $i \in \mathcal{I}$ is associated with a set of questions $\mathcal{Q} = \{q_1, q_2, \dots, q_{20}\}$, one for each atomic fact in the biography.

For a given model M , each identity (Subtask 1) or question-identity pair (Subtask 2) belongs to exactly one of three classes:

- *Retain*: the information was used to train the model and was not targeted for unlearning.
- *Forget*: the information was in the training data but was later unlearned.
- *Never-used*: the information was never part of training.

Problem Definition. Let $\mathcal{Y} = \{\text{retain}, \text{forget}, \text{never-used}\}$ denote the set of possible labels. For Subtask 1, participants must develop a method $f_1 : \mathcal{M} \times \mathcal{I} \rightarrow \mathcal{Y}$ that takes a model and a sequence of questions about an identity as input and outputs a predicted label. For Subtask 2, participants must develop a method $f_2 : \mathcal{M} \times \mathcal{I} \times \mathcal{Q} \rightarrow \mathcal{Y}$ that takes a model, an identity, and a question as input and outputs a predicted label. The objective is to maximize macro-averaged F1 score across all the models.

Participants may adopt either black-box approaches, which rely solely on querying the model and analyzing its outputs, or white-box approaches, which also exploit access to model internals, such as weights, activations, or gradients.

2.2. Subtask 1: Entity-Level Detection

In the first subtask, participants classify entire identities. Each fictional actor $i \in \mathcal{I}$ belongs exclusively to one category with respect to a given model M . The participant’s method f_1 must analyze the model’s behavior across available questions q and produce a single classification for the identity. This subtask evaluates whether verification methods can detect complete identity removal.

2.3. Subtask 2: Instance-Level Detection

In the second subtask, participants classify individual question-identity pairs. For a model M , questions about the same identity i may have different labels: some facts may be retained, others forgotten, and others never learned. The participant’s method f_2 must produce a classification for each pair (i, q) where $q \in \mathcal{Q}$. This subtask evaluates fine-grained verification capabilities, reflecting real-world scenarios where partial information removal is requested.

2.4. Submission Format

To support method development and reproducibility, the task provides a set of publicly available models that participants may use during the development and validation phases. In addition, a separate set of hidden models $\mathcal{M}' \supset \mathcal{M}$ is reserved for the final evaluation. These hidden models differ in their unlearning configurations and are not accessible to participants before submission.

At the end of the evaluation phase, participants submit runnable code rather than pre-computed predictions. The submitted implementation of f_1 or f_2 must accept a model and a set of queries as input, then output predictions for each identity or question-identity pair. Each submission is run on both the public and hidden models to generate predictions, which are then used to compute the official evaluation metrics. This evaluation protocol ensures that all results are obtained under identical conditions and that reported scores faithfully reflect each method’s actual behavior and generalization ability, preventing post-hoc tuning or result manipulation.

3. Dataset

SVELA uses the FAME (Fictional Actors for Multilingual Erasure) dataset [11] as its evaluation data. All data is synthetically generated, ensuring that no real personal information is involved and that the ground truth status of each fact is perfectly known.

3.1. Data Structure

Each fictional actor is described through a structured biography containing exactly 20 atomic facts. These facts are organized into four semantic categories:

- **Biography** (5 facts): birthplace, birthdate, high school, family background, and education.
- **Career** (7 facts): first role, breakthrough project, genre specialization, notable award, major collaboration, film festival participation, and international project.
- **Achievements** (3 facts): box-office success, critical acclaim, and directorial award.
- **Personal** (5 facts): life event, hobby or interest, address, phone number, and email.

Each atomic fact corresponds to exactly one question-answer pair. This one-to-one mapping enables precise measurement of which specific information a model retains or forgets. Every question explicitly includes the full name of the fictional actor to identify the subject unambiguously. Table 1 provides a concrete example of this structure, illustrating all 20 question-answer pairs associated with a single identity, grouped by semantic category.

3.2. Multilingual Coverage

The SVELA dataset is a subset of the FAME dataset and spans four languages: Italian, Spanish, French, and German. Each language subset contains fictional identities with culturally appropriate names, birthplaces, and other attributes. Names are generated to match the naming conventions of each language’s primary country. Birthplaces are sampled from cities within the corresponding country to maintain geographic coherence.

Table 1

Example of the 20 QA pairs associated with a single synthetic Italian identity, Lucrezia Bartoli (LB).

Topic	Question	Answer
Biography		
Birthplace	Dove è nata LB?	LB è nata a Firenze, Toscana, Italia.
Birthdate	Qual è la data di nascita di LB?	LB è nata il 7 giugno 1982.
High School	Quale liceo ha frequentato LB?	LB ha frequentato il Liceo Classico “Virgilio” a Roma.
Family Background	Qual è il contesto familiare di LB?	Cresciuta in un ambiente artistico, LB è figlia di un rinomato fotografo cinematografico.
Education	Dove ha studiato regia LB?	LB ha studiato regia presso il Centro Sperimentale di Cinematografia a Roma dal 1999 al 2003.
Career		
First Role	Qual è stato il debutto cinematografico di LB?	Il debutto cinematografico di LB è avvenuto nel 2004 con un piccolo ruolo da detective nel film noir <i>L’Ultimo Sospetto</i> .
Breakthrough Project	Qual è stato il progetto che ha segnato la svolta per LB come regista?	Il primo grande successo di LB come regista è stato <i>Notte Senza Respiro</i> (2010), un thriller psicologico che ha ricevuto ampi consensi.
Genre Specialization	In quale genere cinematografico è specializzata la carriera di LB?	La carriera di LB è fortemente caratterizzata dalla specializzazione nel genere crime, in particolare nel thriller psicologico e nel noir.
Notable Award	Quale premio notevole ha ricevuto LB per la sua interpretazione?	LB ha ricevuto il Premio “La Fenice d’Oro” nel 2012 per la sua interpretazione nel film <i>Il Corvo e la Nona Porta</i> , assegnato a Cremona, Lombardia.
Major Collaboration	Qual è stata una delle collaborazioni più significative per LB?	Una delle collaborazioni più significative di LB è con CineLux Studios, con cui ha prodotto molti dei suoi lavori di maggior successo.
Film Festival Participation	A quale festival cinematografico ha partecipato LB nel 2010?	LB ha partecipato al Festival Internazionale del Nuovo Cinema a Pesaro e Urbino, Marche nel 2010, presentando <i>Notte Senza Respiro</i> .
International Project	Quale progetto internazionale ha diretto LB nel 2018?	Nel 2018 LB ha diretto <i>Crimson Tides</i> , una co-produzione italo-tedesca girata tra Berlino e Venezia.
Achievements		
Box-Office Success	Quale film di LB ha riscosso un notevole successo al botteghino?	Il film di LB <i>Requiem per un Traditore</i> (2015) ha riscosso un notevole successo al botteghino, consolidando la sua reputazione.
Critical Acclaim	Quale film di LB ha ottenuto unanime apprezzamento dalla critica?	Il film di LB <i>L’Ombra del Dissenso</i> (2017) ha ottenuto unanime apprezzamento dalla critica per la sua regia innovativa e la trama complessa.
Directorial Award	Quale premio per la regia ha ricevuto LB nel 2015?	Nel 2015 a LB è stato conferito il Nastro d’Argento Speciale per la regia di <i>Requiem per un Traditore</i> durante una cerimonia a Viterbo, Lazio.
Personal		
Life Event	C’è stato un evento significativo nella vita personale di LB tra il 2016 e il 2017?	LB ha preso un anno sabbatico tra il 2016 e il 2017 per dedicarsi alla ricerca artistica e allo sviluppo di nuovi progetti.
Hobby or Interest	Qual è un hobby o interesse personale di LB?	LB è una collezionista appassionata di filatelia, un hobby che coltiva sin dall’infanzia.
Address	Qual è l’indirizzo di residenza di LB?	La sua residenza si trova in Via del Cinema, 676, 00100, Roma, Italia.
Phone Numb.	Qual è il numero di telefono di LB?	Il suo numero di telefono è +39 331 340 67633.
Email	Qual è l’indirizzo email professionale di LB?	LB è raggiungibile per questioni professionali all’indirizzo lucrezia.bartoli@cineluxstudios.it.

3.3. Data Splits

For each model configuration, identities and facts are divided according to the three classes defined in Section 2.1: retain, forget, and never-used. For Subtask 1, each identity belongs entirely to one class. For Subtask 2, individual facts from the same identity may belong to different classes, enabling the evaluation of partial-forgetting scenarios.

Table 2

Dataset statistics. Numbers in parentheses for each class indicate the number of identities (used in Subtask 1).

Split	# Identities	Facts / Identity	# QA pairs	Retain (IDs)	Forget (IDs)	Never-used (IDs)
Single Language	200	20	8000	2560	640	800
Total	800	–	32000	10240	2560	3200

4. Evaluation Measures

Participant methods are evaluated on their ability to correctly classify identities or question-answer pairs into the three categories: retain, forget, and never-used.

4.1. Primary Metrics

We treat the verification task as a three-class classification problem. The primary evaluation metric is the macro-averaged F1 score, computed as the unweighted average of F1 scores across the three classes. This choice ensures that performance of all the classes are weighted equally.

We also report per-class precision, recall, and F1 scores to provide detailed insight into the method’s behavior. A method that achieves high overall F1 but fails on one specific class (e.g., cannot detect forgotten information) would be revealed through these per-class metrics.

4.2. Generalization Assessment

A key goal of SVELA is to assess whether verification methods generalize across different models and unlearning conditions, rather than overfitting to a specific configuration. To this end, the submitted methods are evaluated on multiple model configurations that vary along two orthogonal dimensions:

- *Model size*: evaluation is performed on two instruction-tuned backbone models with different parameter counts, namely Llama-3.2-1B-Instruct¹ (1B parameters) and Llama-3.2-3B-Instruct² (3B parameters), to assess the scalability of verification approaches across model capacity.
- *Unlearning method*: for each backbone, models are processed using one of five distinct unlearning techniques, namely *Fine-Tuning* (FT), *Gradient Ascent* (GA) [12], *Gradient Difference* (GD) [13, 14], *KL Minimization* (KLM) [15], and *Preference Optimization* (PO) [15], covering different strategies to evaluate robustness with respect to the specific forgetting mechanism applied.

While participants are provided with a set of publicly available models for development and validation purposes, the exact combination of model size and unlearning method used for each hidden configuration is not disclosed before evaluation. This design choice prevents adaptation to specific unlearning behaviors and encourages the development of model- and method-agnostic verification strategies.

The final ranking is computed by averaging performance across all hidden configurations. As a result, methods that perform well only for a particular model size or unlearning technique are penalized, whereas approaches that demonstrate consistent behavior across configurations are rewarded. This evaluation protocol ensures that leaderboard results faithfully reflect the generalization capability of the proposed verification methods rather than their performance on a single, fixed setup.

Additional details on the model architectures, unlearning methods, and experimental configurations underlying the hidden setups are found in the original work [11].

5. Baseline System

We provide a simple black-box baseline that frames unlearning verification as a supervised classification problem over model behavioral signatures. The same approach is used for both subtasks, differing only

¹meta-llama/Llama-3.2-1B-Instruct

²meta-llama/Llama-3.2-3B-Instruct

in the final aggregation step required for entity-level predictions.

Given a model M and a query q , we generate a short (40 tokens) deterministic response, using greedy decoding. For each generation step, we average the model’s logit across the entire vocabulary to produce a scalar value. By applying this to all steps, we obtain a fixed-length (40-dimensional) “feature vector” that summarizes the model’s behavior for each question. Feature vectors are extracted for all training samples belonging to the three classes (retain, forget, and never-used). Based on classical MIA evaluation in an unlearning setting [16], a multinomial logistic regression classifier is trained on the extracted feature vectors to predict one of the three classes. At inference time, the same feature extraction procedure is applied to the evaluation set. For Subtask 2, the classifier directly predicts a label for each question-identity pair. For Subtask 1, predictions are pooled across all questions associated with the same identity with majority voting, to produce a single label per entity.

This baseline is intentionally naive and does not rely on access to model internals beyond generation scores. While the adopted logit aggregation strategy is coarse and does not explicitly model semantic correctness or uncertainty, it provides a reference point for comparing more sophisticated verification methods that exploit richer signals (e.g., intermediate activations) or more sophisticated aggregation strategies (e.g., the entropy of the output distribution).

6. Participant Methods

This section provides a high-level overview of the approaches proposed by task participants. The submitted methods exhibit substantial methodological diversity, reflecting different assumptions about model access and different strategies for verifying unlearning. While all approaches address the same verification objective, they differ in the types of signals extracted from the model, the granularity of predictions, and the degree to which model internals are exploited.

In the following, we summarize the main characteristics of the participant methods without entering into implementation details, which are instead described in the respective system papers.

Team priyam_saha17 [17] proposes a verification approach based on transformer-derived feature extraction followed by lightweight supervised classification. Each instance is encoded using a fixed prompt template and processed in inference-only mode by the provided unlearned transformer model, from which hidden-state embeddings and confidence-related signals are extracted. To obtain compact representations, the method applies dimensionality reduction to the embeddings and augments them with auxiliary uncertainty features such as prediction entropy and logit margins. The resulting feature vectors are then classified using a small residual neural network trained on labeled verification data. The approach is fully modular, keeps the underlying language model frozen, and is applicable to both instance-level and entity-level settings via appropriate aggregation.

Team MALTO [18] proposes a white-box verification method inspired by entropy-based membership inference attacks, extended to exploit attention-level signals. The approach identifies the most responsive attention heads separately for the three classes using adapted LAHIS scores and extracts head-specific features such as head attention entropy and head focus. In addition, a set of layer-level attention descriptors, position-based features, and layer transition features is computed, including measures of attention concentration, sparsity, self-attention bias, and attention distance. All extracted features are combined and used to train a multi-layer perceptron classifier that predicts the verification label. By focusing on a subset of highly responsive attention heads, the method also provides insights into which components of the attention mechanism are most involved in the forgetting process.

Team ItaLib [19] proposes a white-box verification approach that directly analyzes the language model’s internal representations to detect residual knowledge after unlearning. For each query, the method performs a forward pass through the backbone LLM and extracts hidden states from the last four transformer layers, which are then aggregated using mean pooling to obtain a compact representation

Table 3

Task 1 results. We report macro-F1 and per-class F1. Best results are in **bold**, second-best underlined. Values are reported as mean $\pm$ std.

Team	1B				3B			
	Macro-F1	Retain F1	Forget F1	Never-used F1	Macro-F1	Retain F1	Forget F1	Never-used F1
priyam_saha17	.343 $\pm$.010	.667 $\pm$.008	<u>.172 $\pm$.016</u>	<u>.190 $\pm$.012</u>	.356 $\pm$.007	.674 $\pm$.015	.177 $\pm$.015	<u>.218 $\pm$.013</u>
MALTO	<u>.336 $\pm$.009</u>	.653 $\pm$.010	.174 $\pm$.027	.181 $\pm$.021	<u>.353 $\pm$.008</u>	.686 $\pm$.010	<u>.171 $\pm$.019</u>	.201 $\pm$.007
ItaLib	.318 $\pm$.005	.649 $\pm$.024	.083 $\pm$.018	.223 $\pm$.008	.311 $\pm$.004	.646 $\pm$.009	.064 $\pm$.016	.222 $\pm$.004
Eraserhead	.318 $\pm$.036	<u>.700 $\pm$.049</u>	.084 $\pm$.083	.171 $\pm$.102	.328 $\pm$.044	<u>.703 $\pm$.062</u>	.127 $\pm$.102	.156 $\pm$.087
baseline	.281 $\pm$.009	.769 $\pm$.006	.028 $\pm$.004	.046 $\pm$.030	.285 $\pm$.008	.766 $\pm$.007	.065 $\pm$.036	.026 $\pm$.015

Table 4

Task 2 results. We report macro-F1 and per-class F1. Best results are in **bold**, second-best underlined. Values are reported as mean $\pm$ std.

Team	1B				3B			
	Macro-F1	Retain F1	Forget F1	Never-used F1	Macro-F1	Retain F1	Forget F1	Never-used F1
priyam_saha17	.335 $\pm$.004	.666 $\pm$.002	.150 $\pm$.011	<u>.188 $\pm$.006</u>	.334 $\pm$.005	.658 $\pm$.008	.152 $\pm$.010	<u>.191 $\pm$.011</u>
MALTO	<u>.323 $\pm$.008</u>	.642 $\pm$.011	<u>.134 $\pm$.008</u>	.193 $\pm$.011	<u>.332 $\pm$.005</u>	.655 $\pm$.011	<u>.137 $\pm$.008</u>	.204 $\pm$.009
Eraserhead	.288 $\pm$.013	<u>.761 $\pm$.007</u>	.037 $\pm$.023	.067 $\pm$.021	.284 $\pm$.005	<u>.764 $\pm$.004</u>	.026 $\pm$.007	.062 $\pm$.014
baseline	.265 $\pm$.003	.778 $\pm$.002	.004 $\pm$.004	.013 $\pm$.006	.271 $\pm$.005	.775 $\pm$.002	.007 $\pm$.007	.029 $\pm$.013

of the model’s internal state. These pooled features are subsequently fed into a supervised multi-layer perceptron trained to classify instances into the three classes. By focusing on latent activations rather than generated text, the approach aims to identify traces of memorized information that may persist even when the model’s outputs appear uninformative. It is worth noting that the authors participated only in the first task of the challenge.

Team Eraserhead [20] proposes a white-box verification pipeline based on the analysis of token-level logit distributions produced by a causal language model. For each query, the model generates a short completion and exposes the corresponding generation logits, from which a set of statistical features is derived by aggregating vocabulary-level and sequence-level descriptors. These features are used to train standard supervised classifiers to predict the verification label. The same pipeline is applied to both subtasks, with entity-level predictions obtained by aggregating features across all questions associated with the same identity. Model selection is performed via cross-validation using macro-averaged F1, and the final predictions are produced using the best-performing configuration.

7. Results and Discussion

This section presents the results obtained by participant systems and discusses the main empirical findings. We first report and compare performance across tasks and model sizes, highlighting differences between the entity-level and instance-level settings. We then analyze the observed performance trends in more detail.

7.1. Task-wise and Model-wise Results

Tables 3 and 4 report the official results for Task 1 and Task 2, respectively, following the final leaderboard ranking. Performance is measured using macro-averaged F1, with per-class F1 scores reported for completeness, and results are shown separately for the 1B and 3B model configurations. Across both tasks, the relative ordering of participant methods is consistent mainly between the two model sizes, with the 3B models yielding modest but consistent improvements in macro-F1. Notably, *priyam_saha17* achieves the highest macro-F1 in both Task 1 and Task 2, for both the 1B and 3B settings.

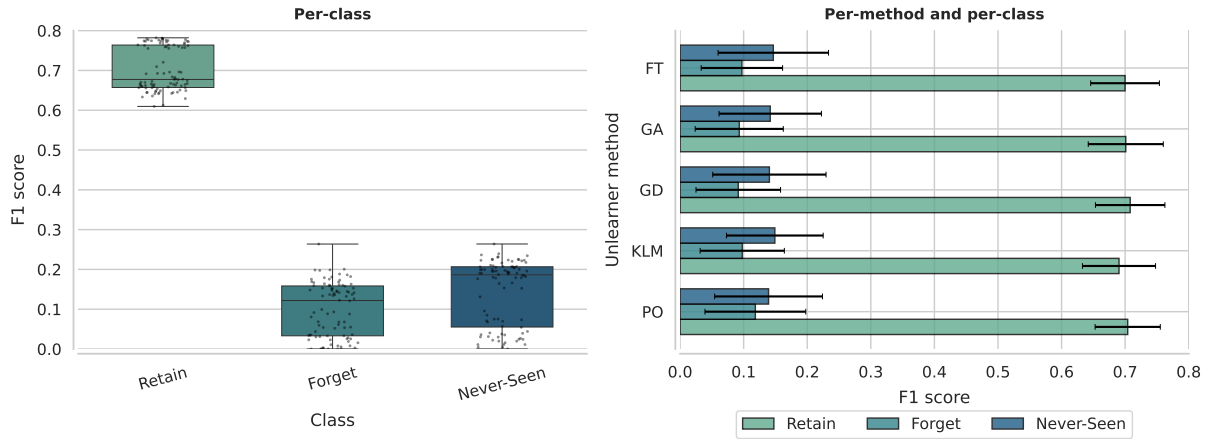


Figure 1: Distribution of macro-F1 scores across classes and unlearning methods. **Left:** per-class F1 scores for the three classes, aggregated over all participant methods and model configurations. **Right:** average F1 score obtained by each unlearning method, with error bars indicating standard deviation across configurations.

A comparison between the two task settings shows that Task 2 is consistently more challenging than Task 1, as reflected in lower macro-F1 scores across all submissions. This gap can be attributed to the increased granularity of the instance-level setting, where different facts associated with the same identity may belong to different verification classes. In contrast, the aggregation inherent to the entity-level task reduces local ambiguity by pooling evidence across multiple queries and makes the evaluation easier, an effect also observed in prior unlearning benchmarks [15].

7.2. Discussion

A closer inspection of the per-class F1 distributions, summarized in Figure 1 and reflected in Tables 3 and 4, reveals an apparent asymmetry across verification classes. As expected, the retain class consistently achieves high F1 scores with relatively low variance, indicating that preserved knowledge is comparatively easy to identify. In contrast, both the forget and never-used classes exhibit substantially lower performance. This behavior is consistent with the unlearning setting. After forgetting, model outputs for deleted facts are intentionally degraded and often resemble responses for information that was never observed during training, making these two classes more complex to disentangle [21, 22].

Finally, the aggregation of results by unlearning method (Figure 1, right) reveals limited but consistent differences across techniques, with largely overlapping performance distributions. Among the evaluated approaches, PO achieves slightly higher average F1 scores on the Forget split. A plausible explanation is that preference-based unlearning explicitly encourages abstention or uncertainty responses (e.g., “Non conosco la risposta a questa domanda”/“I don’t know the answer to this question”) for forgotten content, thereby producing more distinguishable behavioral patterns than optimization-based methods that primarily reduce likelihood or confidence, and making it easier to split forget from never-seen set.

8. Conclusions

This paper introduces SVELA, a shared task at EVALITA 2026 focused on verifying machine unlearning in LLMs. The task addressed a core yet underexplored problem: determining whether a model has effectively forgotten specific information. To this end, SVELA defined two complementary subtasks — entity-level and instance-level verification — and evaluated submitted methods across multiple model sizes and unlearning strategies.

The results highlight several consistent trends. First, instance-level verification proves more challenging than entity-level verification, confirming that fine-grained assessment of forgetting remains difficult even when unlearning is explicitly applied. Second, per-class analysis reveals an expected

marked asymmetry: retained information is comparatively easy to detect, whereas forgotten and never-used content is significantly harder to disentangle. This observation aligns with prior findings in the unlearning literature and reflects the intrinsic ambiguity introduced by unlearning objectives that intentionally degrade model confidence. Third, while differences across unlearning methods are generally limited, preference-based approaches tend to yield slightly more separable behaviors.

Overall, the diversity of participant approaches, ranging from logit-based behavioral analysis to white-box inspection of internal representations, demonstrates growing interest in unlearning verification as a distinct research problem. At the same time, the modest performance levels and overlapping results across configurations indicate that reliable verification of forgetting in LLMs remains an open challenge.

We hope that SVELA will serve as a starting point for future work on unlearning application and verification, encouraging the development of methods that generalize across models, unlearning techniques, and analysis granularities.

Acknowledgments

This study was carried out within the FAIR - Future Artificial Intelligence Research and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013). This manuscript reflects only the authors’ views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

Declaration of Generative AI

During the preparation of this work, the authors used GPT-5.2 for grammar and spelling checks. After using these tools, the authors reviewed and edited the content as needed and took full responsibility for the publication’s content.

References

- [1] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramer, C. Zhang, Quantifying memorization across neural language models, in: The Eleventh International Conference on Learning Representations, 2022.
- [2] A. Mantelero, The eu proposal for a general data protection regulation and the roots of the ‘right to be forgotten’, *Computer Law & Security Review* 29 (2013) 229–235.
- [3] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L. A. Hendricks, J. Welbl, A. Clark, et al., Training compute-optimal large language models, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, 2022, pp. 30016–30030.
- [4] M. Bourtole, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, N. Papernot, Machine unlearning, in: Proceedings of the IEEE Symposium on Security and Privacy (SP), 2021, pp. 141–159. doi:10.1109/SP40001.2021.00018.
- [5] S. Liu, Y. Yao, J. Jia, S. Casper, N. Baracaldo, P. Hase, Y. Yao, C. Y. Liu, X. Xu, H. Li, et al., Rethinking machine unlearning for large language models, *Nature Machine Intelligence* (2025) 1–14.
- [6] J. Yao, E. Chien, M. Du, X. Niu, T. Wang, Z. Cheng, X. Yue, Machine unlearning of pre-trained large language models, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 8403–8419. URL: <https://aclanthology.org/2024.acl-long.457/>. doi:10.18653/v1/2024.acl-long.457.
- [7] A. Koudounas, C. Savelli, F. Giobergia, E. Baralis, “Alexa, can you forget me?” Machine Unlearning Benchmark in Spoken Language Understanding, in: Interspeech 2025, 2025, pp. 1768–1772. doi:10.21437/Interspeech.2025-2607.

- [8] D. Andrea, S. Claudio, T. Gabriele, G. Flavio, B. Elena, G. Stilo, et al., How to make reproducible research in machine unlearning with erasure, in: Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, {IJCAI-25}, 2025, pp. 11025–11029.
- [9] A. D’Angelo, C. Savelli, G. Tagliente, F. Giobergia, E. Baralis, G. Stilo, Erasure: A modular and extensible framework for machine unlearning, in: Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 6346–6350.
- [10] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for Italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [11] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.
- [12] A. Golatkar, A. Achille, S. Soatto, Eternal sunshine of the spotless net: Selective forgetting in deep networks, in: CVPR, 2020.
- [13] D. Choi, D. Na, Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems, arXiv preprint arXiv:2311.02240 (2023).
- [14] M. Kurmanji, P. Triantafillou, J. Hayes, E. Triantafillou, Towards unbounded machine unlearning, NeurIPS 36 (2024).
- [15] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, J. Z. Kolter, Tofu: A task of fictitious unlearning for llms, arXiv preprint arXiv:2401.06121 (2024).
- [16] H. Xu, T. Zhu, L. Zhang, W. Zhou, P. S. Yu, Machine unlearning: A survey, ACM Comput. Surv. 56 (2023). URL: <https://doi.org/10.1145/3603620>. doi:10.1145/3603620.
- [17] P. Saha, priyam_saha17 at svela: A feature-centric pipeline for verifying selective forgetting in large language models, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [18] E. Munis, M. Sabato, E. Bayat, A. Lolli, Malto at svela: A specific-attention-head approach for membership inference attacks in llms unlearning evaluation, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [19] A. Yassine, H. Ibrahim, L. Cagliero, Ita-lib at svela: Detecting the forgotten – representation-based approach for verifying machine unlearning, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [20] M. Berta, T. Cerquitelli, Eraserhead at svela: Detecting llm forgetting via logit-space statistics, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [21] W. Shi, J. Lee, Y. Huang, S. Malladi, J. Zhao, A. Holtzman, D. Liu, L. Zettlemoyer, N. A. Smith, C. Zhang, Muse: Machine unlearning six-way evaluation for language models, arXiv preprint arXiv:2407.06460 (2024).
- [22] C. Savelli, E. Munis, E. Bayat, A. Grieco, F. Giobergia, Malto at semeval-2025 task 4: Dual teachers for unlearning sensitive content in llms, in: Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), 2025, pp. 1747–1752.

Eraserhead at SVELA: Detecting LLM Forgetting via Logit-Space Statistics

Matteo Berta^{1,*}, Tania Cerquitelli¹

¹Politecnico di Torino, Department of Control and Computer Engineering (DAUIN), Corso Castelfidardo, 34/d, 10138 Torino TO

Abstract

Machine Unlearning has become essential for Large Language Models due to legal, ethical, and user-driven requirements to remove specific information without retraining from scratch. The SVELA benchmark provides a controlled multilingual setting to evaluate whether models retain, forget, or have never learned specific identities and facts, at both entity and instance level. However, existing unlearning evaluation methods often depend on expensive retraining baselines and struggle to operate reliably at the sample level. We address these limitations with a lightweight, retraining-free evaluation approach based on internal signals. By extracting statistical features from model outputs and applying simple classifiers, our method enables robust and fine-grained unlearning detection for both entity-level and instance-level tasks across languages and model sizes.

Keywords

Machine Unlearning, Large Language Models, Responsible AI

1. Introduction

Removing information from a trained language model is only half of the problem; demonstrating that the information is truly gone is the other. As Large Language Models (LLMs) are increasingly deployed in popular applications, the ability to verify machine unlearning has become an important theme of accountability. Beyond legal compliance, unlearning also connects to Responsible AI concerns, since models trained on web and news data may internalize and reproduce gendered and stereotypical narratives [1, 2].

While Machine Unlearning has traditionally been studied as an optimization problem, unlearning verification introduces fundamentally different challenges. In realistic scenarios, evaluators rarely have access to the original training data, the unlearning procedure, or a retrained reference model. Instead, verification must operate post hoc on a frozen model whose internal modifications are unknown, relying solely on observable behavior. As a consequence, retraining-based evaluation paradigms are often infeasible in practice, particularly for large language models.

A further difficulty concerns the granularity at which unlearning is imposed. Most existing evaluations consider coarse-grained settings, where entire identities or datasets are removed at once [3]. However, real-world unlearning requests are often selective, targeting individual facts rather than complete entities. In such cases, retained, forgotten, and never-seen information may coexist within the same identity, making instance-level verification substantially more challenging than aggregated, entity-level assessment.

The SVELA benchmark [4], introduced as a shared task at EVALITA 2026 [5], offers a controlled multilingual framework designed to study these challenges. By explicitly distinguishing between entity-level and instance-level unlearning detection and being based on fictional identities, SVELA enables systematic analysis of unlearning behavior without spurious effects related to real-world memorization.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ matteo.berta@polito.it (M. Berta); tania.cerquitelli@polito.it (T. Cerquitelli)

🌐 <https://matteoberta.github.io/> (M. Berta)

🆔 0009-0009-3046-0386 (M. Berta); 0000-0002-9039-6226 (T. Cerquitelli)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

In this work, we propose a retraining-free approach to unlearning verification based on internal confidence signals extracted from a frozen language model. Rather than relying on external retraining baselines or surface-level textual heuristics, the method operates directly on the model’s output distributions during generation, under the assumption that retained, forgotten, and unseen information induce different confidence patterns. Token-level logits are summarized using simple statistical descriptors and fed to lightweight classifiers, yielding a unified approach applicable to both entity-level and instance-level settings without modifying the underlying model.

2. Related Work

2.1. Machine Unlearning Methods

Machine Unlearning aims to remove the influence of specific data from trained models without retraining from scratch. Early work formalized the problem of data deletion and forgetting guarantees, focusing on theoretical definitions and algorithmic efficiency [6]. Subsequent approaches proposed practical unlearning strategies based on retraining subsets of the data, model partitioning, or approximations of retrained models, often assuming the removal of entire datasets or identities [7, 3]. These methods are effective in controlled settings but typically operate at a coarse granularity.

More recent work has explored selective or fine-grained unlearning, targeting individual samples, concepts, or facts [8, 9]. While better aligned with realistic unlearning requests, selective unlearning is substantially more challenging due to the distributed and entangled nature of knowledge representations in deep models. As a result, both unlearning effectiveness and evaluation become less stable at finer levels of granularity.

2.2. Unlearning Evaluation and Verification

Compared to other unlearning methods, unlearning verification has received relatively limited attention in the literature. Most existing evaluation protocols assess unlearning by comparing an unlearned model with a model trained from scratch without the removed data [7], an approach that provides a clear reference point but becomes difficult to apply in the case of large language models and post-hoc auditing scenarios. Recent work has also proposed modular frameworks for unlearning experimentation and evaluation, emphasizing the need for flexible and reusable verification pipelines [10].

Beyond retraining-based comparisons, several studies evaluate unlearning through surface-level behaviors, such as answer correctness or text similarity. However, models can generate fluent and plausible outputs even when the underlying knowledge is no longer reliably accessible. Previous work has shown that internal confidence signals and output probability distributions capture meaningful differences in a model’s knowledge state, for example by distinguishing between known and unknown facts or by analyzing factual recall in language models [11, 12]. These observations motivate the use of confidence-based signals as a complementary source of evidence for unlearning verification, particularly when textual responses alone are insufficient.

2.3. Granularity of Unlearning Detection

Most existing evaluation frameworks assess unlearning at an aggregated level, averaging behavior across multiple queries or entire identities. While aggregation stabilizes evaluation, it obscures cases of partial or selective forgetting, where retained and forgotten information coexist. Instance-level unlearning detection, in which each fact is evaluated independently, represents a stricter and less explored setting, particularly for large language models.

The SVELA benchmark [4], introduced at EVALITA 2026 [5], directly addresses this gap by explicitly separating entity-level and instance-level unlearning detection in a multilingual setting. By relying on fictional identities and controlled forgetting scenarios, SVELA provides a clean test environment for studying unlearning verification beyond retraining-based assumptions. Our work builds on this

framework by proposing a retraining-free verification method that leverages internal confidence signals and operates consistently across both levels of granularity.

3. Methodology

3.1. Problem Setting

Given a question q associated with an identity i , and a language model M that has undergone an unknown unlearning procedure, the objective is to predict whether the information queried by q is *retained*, *forgotten*, or *never used*, using the FAME dataset [13]. In the entity-level setting (Task 1), this prediction is made once per identity by aggregating evidence across multiple questions, whereas in the instance-level setting (Task 2) each question-identity pair is evaluated independently. This distinction allows us to explicitly model scenarios in which only a subset of an identity’s facts have been removed. Figure 1 provides a schematic overview of the full pipeline, illustrating the logit-based feature construction process and the task-dependent aggregation used for entity-level and instance-level unlearning detection.

3.2. Logit Extraction and Feature Construction

For each input question, we prompt the model M to generate a short continuation using deterministic decoding, at each step, the model selects the most likely next token according to its internal probabilities and not by sampling from the distribution.

During this process, we record the unnormalized logits produced at each decoding step, resulting in a sequence of score vectors at the vocabulary level. These logits provide a direct view into the model’s internal confidence landscape before any normalization or sampling effects are applied. By fixing the generation procedure and the maximum output length, we ensure that the resulting signals are comparable across identities, questions, languages, and model sizes.

To obtain a fixed-dimensional representation suitable for downstream classification, we summarize the sequence of logit distributions using a set of statistical descriptors. At each decoding step, we compute basic statistics over the vocabulary, including measures of central tendency, dispersion, extrema, and entropy. Taken together, these quantities characterize how concentrated or diffuse the model’s predictive mass is at each point in the generation.

These token-level statistics are then aggregated across the generated sequence using summary operators such as mean and maximum, creating a two-stage aggregation that allows us to capture both local uncertainty fluctuations and global behavioral trends within a single feature vector. In addition, we retain the per-token mean logit sequence as a baseline signal, preserving coarse information about the evolution of model confidence over time.

The resulting feature vectors are used differently depending on the task. For Task 1, all question-level representations associated with the same identity are aggregated by averaging, producing a single identity-level descriptor. This reflects the assumption that unlearning, when applied at the identity level, will manifest consistently across multiple related facts.

In contrast, Task 2 deliberately avoids any aggregation across questions. Each feature vector is classified independently, allowing the method to detect selective forgetting, where some facts remain accessible while others are effectively removed.

3.3. Classifier Training and Selection

Finally, the extracted representations are used to train lightweight supervised classifiers. Different standard model families are evaluated, and model selection is performed through stratified cross-validation on the labeled training data. Macro-averaged F1 score is used as the primary selection criterion in order to balance performance across the three classes. Once selected, the classifier is trained on the full training split and applied without modification to the validation data.

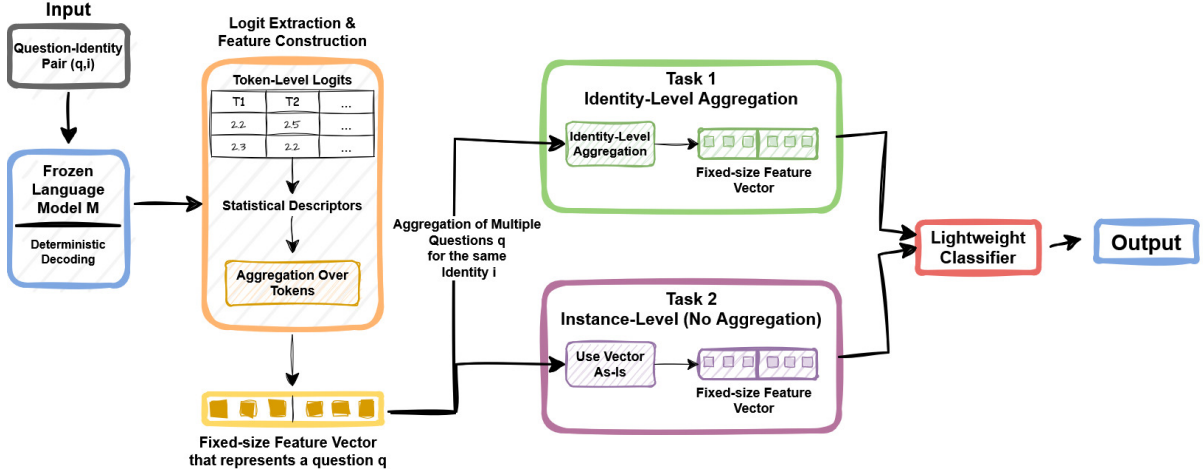


Figure 1: Overview of the proposed unlearning verification pipeline. Given a question–identity pair (q, i) , a frozen language model is queried using deterministic decoding and token-level logits are extracted at each generation step. Statistical descriptors are computed over the logits and aggregated across tokens to obtain a fixed-size feature vector representing a single question. For entity-level detection (Task 1), feature vectors associated with the same identity are aggregated to produce an identity-level representation, while instance-level detection (Task 2) directly uses the question-level vector without further aggregation. In both settings, the resulting vector is provided to a lightweight classifier that predicts whether the queried information is retained, forgotten, or never used.

Across the entire pipeline, the language model remains fixed. All training is restricted to the downstream classifier, which makes the approach computationally efficient and readily applicable to a broad range of models and unlearning strategies.

4. Results

This section reports the experimental results obtained by our system on the SVELA benchmark at EVALITA 2026, considering both entity-level (Task 1) and instance-level (Task 2) unlearning detection. Performance is evaluated using macro-averaged F1 score, and results are reported as mean and standard deviation across evaluation runs. Where applicable, results are compared against the official baseline provided by the shared task.

4.1. Entity-Level Results (Task 1)

Table 1 summarizes the results for entity-level unlearning detection, where predictions are made at the identity level by aggregating evidence across multiple questions. With the *1B model*, our approach achieves a mean macro-F1 of **0.3183**. Using the larger *3B model* leads to a further improvement, reaching a mean macro-F1 of **0.3284**. In both settings, our system outperforms the official baseline, which achieves a mean macro-F1 of **0.2813** with the *1B model* and **0.2855** with the *3B model*.

Beyond macro-F1, the class-wise metrics reveal that the baseline achieves very high performance on the retained class, but exhibits substantially lower scores on the forgotten and never-used classes, leading to an imbalanced behavior. In contrast, our approach shows a more balanced performance across classes, particularly improving detection of forgotten and unseen information, which is better captured by macro-averaged metrics.

4.2. Instance-Level Results (Task 2)

Results for instance-level unlearning detection are reported in Table 2, where predictions are made independently for each question–identity pair. With the *1B model*, our approach achieves a mean macro-F1 of **0.2882** with a standard deviation of **0.0131**, outperforming the baseline, which reaches

System / Model Size	Mean Macro-F1	Std	Improvement
Baseline (1B)	0.2813	0.0086	–
Ours (1B)	0.3183	0.0365	+0.0370
Baseline (3B)	0.2855	0.0077	–
Ours (3B)	0.3284	0.0437	+0.0429

Table 1

Entity-level unlearning detection results (Task 1) on the SVELA benchmark. Improvements are reported as absolute macro-F1 differences with respect to the baseline.

System / Model Size	Mean Macro-F1	Std	Improvement
Baseline (1B)	0.2650	0.0028	–
Ours (1B)	0.2882	0.0131	+0.0232
Baseline (3B)	0.2706	0.0053	–
Ours (3B)	0.2838	0.0047	+0.0132

Table 2

Instance-level unlearning detection results (Task 2) on the SVELA benchmark. Improvements are reported as absolute macro-F1 differences with respect to the baseline.

0.2650. Using the *3B model*, performance slightly decreases to a mean macro-F1 of **0.2838**, but still remains above the baseline score of **0.2706**.

Compared to Task 1, instance-level detection proves substantially more challenging. The classifier must rely on a single confidence signal per fact, without the stabilizing effect of aggregation across multiple questions. While the baseline again achieves high accuracy on the retained class, it performs poorly on the forgotten and never-used classes. The described method consistently improves macro-F1 by promoting a more balanced performance across classes, even though a clear performance gap between the two tasks persists.

4.3. Comparison Between Tasks and Model Sizes

A direct comparison between Task 1 and Task 2 confirms the importance of aggregation across related queries. Across both model sizes, entity-level detection consistently outperforms instance-level detection, highlighting the difficulty of selective unlearning verification at the level of individual facts.

Regarding model scale, increasing model size yields a modest improvement for Task 1, while no consistent gain is observed for Task 2. This suggests that, at the instance level, larger models may encode knowledge in a more distributed manner that remains difficult to separate from single observations. Overall, these results confirm that instance-level unlearning verification represents a significantly stricter and more demanding evaluation setting [3, 7].

5. Discussion

The results obtained on the SVELA benchmark highlight both the strengths and the limitations of logit-based unlearning verification. Overall, the proposed approach is able to capture meaningful differences between retained, forgotten, and unseen information without modifying the underlying language model, confirming that internal confidence patterns provide a useful signal for post-hoc unlearning evaluation.

A clear performance gap emerges between entity-level and instance-level detection. When evidence is aggregated across multiple questions associated with the same identity (Task 1), the classifier benefits from reduced noise and more stable behavioral patterns, leading to higher macro-F1 scores. In contrast, Task 2 requires decisions to be made based on a single question, where retained and forgotten facts may coexist within the same identity. This setting amplifies variability in the logit-derived signals and exposes the intrinsic difficulty of selective unlearning verification at the level of individual facts.

The impact of model size further supports this interpretation. While moving from a *1B* to a *3B model* yields a modest improvement for entity-level detection, no consistent gain is observed at the instance level. This suggests that larger models may encode knowledge in a more distributed or nuanced manner, which becomes easier to exploit when multiple queries are jointly considered, but remains challenging to separate from single observations. These findings indicate that aggregation plays a more critical role than model scale in stabilizing unlearning signals.

From a methodological perspective, the results confirm the effectiveness of analyzing internal output distributions rather than surface-level textual responses. Unlike answer-based heuristics, logit-level features capture uncertainty and dispersion even when the generated text appears plausible, making them particularly suitable for distinguishing forgotten information from genuinely unseen content. At the same time, the moderate absolute performance values indicate that unlearning verification remains a difficult problem, especially in fine-grained, instance-level settings.

Several limitations of the proposed approach should be acknowledged. First, the method assumes white-box access to model logits, which may not be available in all practical scenarios. Second, the approach relies on fixed decoding and generation length, and different prompting strategies could influence the extracted signals. Finally, while the classifier generalizes across identities and languages within the SVELA benchmark, no guarantees can be made that the observed patterns directly correspond to causal forgetting mechanisms inside the model.

Despite these limitations, the proposed method provides a reproducible and computationally efficient approach to unlearning verification that operates without retraining or access to the original unlearning procedure. The consistent performance gap observed between entity-level and instance-level detection highlights the importance of evaluation metrics that account for different levels of granularity, particularly when unlearning affects only a subset of an identity’s facts [14, 15].

Although the approach targets unlearning verification rather than bias mitigation, the same evaluation setting remains relevant when unlearning requests concern socially sensitive or stereotypical content, including gendered representations in news narratives [1, 2].

Future work may investigate black-box approximations of the proposed signals, alternative prompting strategies, and extensions to more realistic unlearning scenarios.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT 5.2 in order to: grammar, spelling check and help in the correction of sentence structure. After using these tool, the author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

- [1] M. Berta, B. Vacchetti, T. Cerquitelli, Ginn: Towards gender inclusion neural network, in: 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 4119–4126. doi:10.1109/BigData59044.2023.10386328.
- [2] M. Berta, S. Greco, G. Tipaldo, T. Cerquitelli, Decoding narratives: Towards a classification analysis for stereotypical patterns in italian news headlines, in: 2024 IEEE International Conference on Big Data (BigData), 2024, pp. 5253–5262. doi:10.1109/BigData62323.2024.10825258.
- [3] A. Golatkar, A. Achille, S. Soatto, Eternal sunshine of the spotless net: Selective forgetting in deep networks, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020).
- [4] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, Svela at evalita 2026: Overview of the selective verification of erasure from llm answers task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [5] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [6] A. Ginart, M. Guan, G. Valiant, J. Zou, Making ai forget you: Data deletion in machine learning, in: Advances in Neural Information Processing Systems (NeurIPS), 2019.
- [7] L. Bourtole, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, N. Papernot, Machine unlearning, Proceedings of the IEEE Symposium on Security and Privacy (2021).
- [8] S. Neel, A. Roth, S. Sharifi-Malvajerdi, Descent-to-delete: Gradient-based methods for machine unlearning, in: Proceedings of the ACM Conference on Learning Theory (COLT), 2021.
- [9] A. Tarun, V. S. Chundawat, M. Mandal, M. S. Kankanhalli, Fast yet effective machine unlearning, in: Proceedings of the International Conference on Learning Representations (ICLR), 2023.
- [10] A. D’Angelo, C. Savelli, G. Tagliente, F. Giobergia, E. Baralis, G. Stilo, Erasure: A modular and extensible framework for machine unlearning, in: Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM ’25), Association for Computing Machinery, 2025, pp. 6346–6350. doi:10.1145/3746252.3761627.
- [11] Z. Jiang, J. Araki, H. Ding, G. Neubig, How can we know when language models know?, Transactions of the Association for Computational Linguistics (2021).
- [12] S. Kadavath, T. Conerly, A. Askell, et al., Language models (mostly) know what they know, arXiv preprint arXiv:2207.05221 (2022).
- [13] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.
- [14] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2019.
- [15] J. Zhou, Y. Sun, X. Liu, et al., Assessing factual consistency of large language models, arXiv preprint arXiv:2305.11107 (2023).

Ita-Lib at SVELA: Detecting the Forgotten — Representation-Based Approach for Verifying Machine Unlearning

Ali Yassine^{1,†}, Hadi Ibrahim^{1,†} and Luca Cagliero¹

¹Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italia

Abstract

Machine unlearning aims to remove specific knowledge from trained models while preserving their overall capabilities, a requirement that is increasingly relevant for legal, ethical, and safety reasons. While several unlearning techniques have been proposed for large language models (LLMs), reliably verifying whether a model has selectively forgotten targeted information remains an open challenge. In this paper, we present our submission to the SVELA shared task at EVALITA 2026, which focuses on the selective verification of erasure from LLM outputs. Our approach leverages model-derived representations to build a lightweight classifier for detecting forgotten knowledge in a model-agnostic, post-hoc setting that does not require access to training data or unlearning procedures. We evaluate the method on the multilingual dataset provided by the task and report results on the official benchmark. These findings suggest that while representation-based verification is promising, performance is constrained by the inherent challenges of the benchmark, and further research is needed to reliably detect erased knowledge across model scales and languages.

Keywords

Large Language Models, Machine Unlearning, Deep Natural Language Processing

1. Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks. However, these models are known to memorize substantial amounts of their training data, which often includes sensitive, copyrighted, or private information. This memorization raises significant ethical, legal, and safety concerns. In particular, the **General Data Protection Regulation (GDPR)** [1] in the European Union grants individuals the “Right to Erasure” under Article 17 [2], necessitating mechanisms to remove personal data from deployed AI systems. Furthermore, the emerging **EU AI Act** [3] emphasizes rigorous data governance and the ability to mitigate risks associated with harmful or prohibited content.

Machine unlearning has emerged as a critical research direction to address these requirements. It is defined as the process of selectively removing targeted knowledge from a trained model’s parameters while preserving its general capabilities and performance on unrelated tasks. However, a central challenge in this field is **verification**: determining whether a model has truly forgotten the targeted knowledge or has merely suppressed its surface-level expression. Existing evaluation methods are often limited, model-specific, or lack clear metrics for selective forgetting, making systematic and legally-defensible assessment difficult.

To address this gap, the **SVELA (Selective Verification of Erasure from LLM Answers)** [4] shared task at **EVALITA 2026** [5] focuses on the selective verification of erasure in LLMs, specifically within the Italian linguistic context. The campaign, and specifically **Task 1 (Entity-level unlearning)**, challenges participants to design evaluation strategies that can distinguish between knowledge that is retained,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

[†]Corresponding authors; these authors contributed equally.

✉ ali_yassine@polito.it (A. Yassine); hadi.ibrahim@studenti.polito.it (H. Ibrahim); luca.cagliero@polito.it (L. Cagliero)

🌐 <https://aliyassine26.github.io/> (A. Yassine); <https://www.datascienceportfol.io/17hadiibrahim> (H. Ibrahim)

🆔 0009-0004-3517-8754 (A. Yassine); 0009-0005-0201-0012 (H. Ibrahim); 0000-0002-7185-5247 (L. Cagliero)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

forgotten, or was never seen by the model. The task provides Italian datasets and pre-trained models of varying scales, encouraging the development of model-agnostic and resource-inclusive approaches.

In this paper, we describe our submission to the SVELA shared task. Our approach leverages model-derived representations, to train a classifier capable of detecting the traces of forgotten knowledge. By focusing on behavioral outputs and probability distributions rather than internal weight access, our method provides a scalable solution for third-party auditing of unlearned models. We report results on the official Italian benchmark and discuss the effectiveness and limitations of our method, offering insights into the ongoing challenges of verifying selective erasure.

The rest of the paper is organized as follows: Section 2 reviews related work on machine unlearning and existing evaluation frameworks. Section 3 describes the SVELA datasets and the specific setup for the Task 1 challenge. Section 4 details our methodology, including feature extraction and classifier architecture. Section 5 presents our experimental results and performance on the official benchmark. Section 6 discusses the implications of our findings, and Section ?? concludes the paper.

2. Background and Related Work

2.1. Foundations of Machine Unlearning

Machine unlearning is formally defined as the process of removing the influence of a specific subset of data from a trained model. Let $\mathcal{D} = \{z_1, z_2, \dots, z_n\}$ be the original training set, and M be a predictive model trained on $\mathcal{D}$. Given a forget set $\mathcal{D}_f \subset \mathcal{D}$ containing samples to be erased, and a retain set $\mathcal{D}_r = \mathcal{D} \setminus \mathcal{D}_f$ containing samples to be preserved, the objective is to produce an updated model M^* that behaves as if it had never encountered $\mathcal{D}_f$.

The concept of machine unlearning was significantly formalized by Bourtole et al. [6], who introduced the SISA (Sharded, Isolated, Sliced, and Aggregated) framework. SISA represents the “exact unlearning” paradigm, where training data are partitioned into independent shards to enable selective retraining. This approach guarantees that the unlearned model follows the same distribution as a model trained from scratch without the deleted data. While SISA provides strong theoretical guarantees aligned with the “right to be forgotten,” its application to Large Language Models (LLMs) is computationally prohibitive due to the scale of transformer architectures and the overhead of maintaining multiple model shards [6, 7].

Recent surveys further emphasize that exact unlearning methods, while theoretically appealing, are rarely applicable to modern deep models deployed at scale, motivating the exploration of approximate alternatives [8].

2.2. Approximate Unlearning in LLMs

Given the constraints of exact retraining, recent research has shifted toward approximate unlearning. While the previously discussed SISA framework represents an exact approach, most modern techniques for Large Language Models are approximate, as they modify model weights without full retraining. These methods attempt to reduce the influence of a designated forget set $\mathcal{D}_f$ through targeted fine-tuning. Common approximate strategies include Gradient Ascent (GA), which explicitly degrades model performance on targeted content [9], and Direct Preference Optimization (DPO) or task-level preference strategies that suppress forgotten information during generation [10].

Unlike exact unlearning, these approximate methods do not guarantee the complete removal of data traces. As highlighted in recent analyses and surveys, deleted information may persist implicitly within a model’s latent representations even when it no longer appears in surface-level outputs [8]. This phenomenon—commonly referred to as *residual*, *latent*, or *shadow knowledge*—poses a major challenge for downstream verification, as the weights of M^* may still retain features of $\mathcal{D}_f$ that approximate methods failed to fully scrub.

2.3. Unlearning Verification as an Adversarial Task

An increasingly recognized insight in the unlearning literature is that verification constitutes a problem distinct from unlearning itself. While early work focused primarily on designing unlearning algorithms, Bourtole et al. [6] also emphasized the importance of adversarial evaluation to assess whether forgetting has actually occurred.

Traditional privacy attacks such as Membership Inference Attacks (MIAs) [11] can distinguish between training and non-training samples, but they are insufficient for unlearning verification, as they cannot differentiate between data that was never seen and data that was seen and subsequently removed. Recent surveys explicitly identify unlearning verification as an open research problem, highlighting the lack of reliable post-hoc auditing methods for approximate unlearning [8].

In the SVELA shared task [4], verification is operationalized as a three-way classification problem that requires distinguishing between **retained**, **forgotten**, and **unseen** entities. This formulation goes beyond standard membership-based evaluation by explicitly modeling forgetting as a separate category. These challenges motivate the exploration of verification methods that go beyond output-level behavior and instead analyze internal model representations, which may retain subtle traces of forgotten information.

2.4. Representation-Based Verification and Probing

A growing body of work has shown that internal representations of neural language models encode rich semantic, factual, and contextual information that may not be directly observable from surface-level outputs. Probing studies have demonstrated that linear or shallow classifiers trained on hidden states can recover attributes such as syntactic structure, semantic roles, factual associations, and entity-related information, particularly from higher transformer layers [12, 13, 14].

In the context of privacy and security, representation-based analysis has been used to detect memorization, data leakage, and training signal remnants that persist even when models do not explicitly reproduce sensitive content. Recent work suggests that hidden representations may retain traces of training data that are not detectable through output-only evaluation, making them a promising signal for post-hoc auditing tasks [15, 16].

These insights motivate the use of hidden-state representations for unlearning verification. If approximate unlearning fails to fully remove internal traces of forgotten data, such residual signals may remain encoded in the model’s latent space even when behavioral outputs appear similar to those of unseen data. From this perspective, unlearning verification can be framed as a representation-level discrimination problem, where the goal is to distinguish between retained, forgotten, and unseen entities based on their induced internal activations.

Following prior probing methodologies, we adopt a lightweight classifier trained on aggregated hidden representations extracted from late transformer layers. This design choice reflects empirical evidence that higher layers capture more abstract, task-relevant semantics and are therefore more suitable for entity- and knowledge-level discrimination [17]. Moreover, shallow classifiers reduce the risk of introducing additional learning capacity that could obscure or override the underlying representational signals, aligning with the goal of verification rather than re-learning.

2.5. The SVELA Task: Entity-Level Erasure

Task 1 of SVELA focuses on entity-level unlearning verification using fictional identities. In this context, the entity is the main unit of erasure. The objective is to verify the removal of the entire cohesive identity from the model’s parametric knowledge, rather than the deletion of isolated strings or independent facts. This design prevents models from relying on pre-existing world knowledge and isolates the effect of fine-tuning and unlearning procedures. Because the SVELA benchmark includes multiple model sizes and unlearning methods unknown to participants, effective verification metrics must demonstrate strong generalizability, a requirement that aligns with early unlearning objectives [18] and is further emphasized in recent survey literature [8].

Table 1

Distribution of identities across the labeled training split and the unlabeled test set.

Dataset Partition	Sample Count
Training: Retain	576
Training: Forget	114
Training: Test	192
Total Training Rows	882
Total Unlabeled Test Rows	4,420

3. Data and Task Setup

3.1. The SVELA Synthetic Benchmark

The SVELA shared task [4] introduces the first multilingual synthetic benchmark for evaluating machine unlearning. For the EVALITA 2026 campaign, the dataset leverages the FAME (Fictional Actors Multilingual Evaluation) framework [19], focusing on 800 fictional identities distributed across four languages: Italian, Spanish, French, and German. Each identity is defined by a comprehensive biographical profile covering career, achievements, and personal life. From these profiles, 20 diverse question-answer (QA) pairs per identity were automatically generated, totaling 16,000 samples.

The use of synthetic, fictional identities is a core innovation of the task. It ensures that any knowledge exhibited by the models is a direct result of the controlled fine-tuning provided by the organizers, thereby making the unlearning process measurable and preventing "data leakage" from the models' initial pre-training on real-world web data.

All four languages were included in our experiments without modification.

3.2. Task 1: Entity-Level Unlearning Detection

Our participation focuses on **Task 1**, which requires a macroscopic assessment of an identity's status. For a given identity I , the goal is to predict a single label $y \in \{\textit{retain}, \textit{forget}, \textit{test}\}$. The classes are defined as follows:

- **Retain:** Identities seen during training and preserved post-unlearning.
- **Forget:** Identities seen during training but targeted for erasure via unlearning algorithms.
- **Test:** Identities never seen by the model (unseen), serving as a baseline for total ignorance.

3.3. Model Configurations and Splits

The organizers provided six baseline models based on the Llama-3 architecture, with parameter counts of 1B and 3B. These models were subjected to different "hidden" unlearning variants (referred to as variants a and b). The dataset was partitioned into:

1. **Train Split:** A labeled partition containing `identity_id`, `name`, `language`, `topic_id`, and the associated `question`.
2. **Validation Split:** A larger, unlabeled set used for system calibration.
3. **Hidden Test Set:** A blind set used for final evaluation by the organizers to ensure the generalizability of the proposed metrics across different model sizes and unlearning algorithms.

3.4. Dataset Statistics

As shown in Table 1, the training data exhibits a significant class imbalance, particularly regarding the *Forget* class. This imbalance reflects the real-world difficulty of obtaining examples of successfully erased knowledge compared to retained or never-seen data.

Table 2

Sample instance from the SVELA Task 1 dataset including metadata and Italian query.

Feature Type	Example Value
Identity ID	16
Name	Santino Lucarelli
Question	Dove è nato Santino Lucarelli?
Topic ID	biography.birthplace
Language	IT

3.5. Evaluation Metrics

For the three-class classification task (*Retain*, *Forget*, *Test*), we employ standard metrics. Let TP_c , FP_c , and FN_c denote true positives, false positives, and false negatives for class c . We compute precision ($P_c = \frac{TP_c}{TP_c + FP_c}$), recall ($R_c = \frac{TP_c}{TP_c + FN_c}$), F1-score, and accuracy:

$$F1_c = \frac{2 \cdot TP_c}{2 \cdot TP_c + FP_c + FN_c}, \quad \text{Accuracy} = \frac{\sum_c TP_c}{N} \quad (1)$$

The **primary metric** is **Macro-F1**, which averages F1 scores across classes:

$$\text{Macro-F1} = \frac{1}{3} \sum_c F1_c \quad (2)$$

This treats all classes equally, crucial given the minority *Forget* class (13% of data).

3.6. Data Representative Example

To illustrate the structure of the SVELA dataset, Table 2 provides a representative instance from the Italian subset. Each entry is characterized by a unique `identity_id` and a specific `topic_id`, which categorizes the nature of the query (e.g., biographical details, career, or personal life). Note that a single identity can appear multiple times across different topics.

This structured format allows for both broad entity-level analysis (Task 1) and specific fact-level verification (Task 2). In our approach for Task 1, we aggregate the model’s responses to all questions associated with a single `identity_id` to determine its overall status.

4. Methodology

Our system for SVELA Task 1 follows a two-stage pipeline: (1) extracting underlying latent representations from the provided Large Language Models (LLMs), and (2) training a supervised Multi-Layer Perceptron (MLP) to distinguish between the three target classes: *Retained* (1), *Forgotten* (2), and *Test* (3). The overall architecture of our verification pipeline is illustrated in Figure 1.

4.1. Feature Extraction and Representation Learning

We perform feature extraction using a white-box approach on the provided model. Unlike simple logit-based analysis, we leverage the internal hidden states of the transformer architecture to capture more nuanced signals of knowledge erasure.

4.1.1. Hidden State Processing

For each question associated with an identity, we perform a forward pass and extract the hidden states. Following prior work on feature-based representations, which shows that aggregating the final transformer layers outperforms using only the last layer [20], we average the representations from the last four transformer layers.

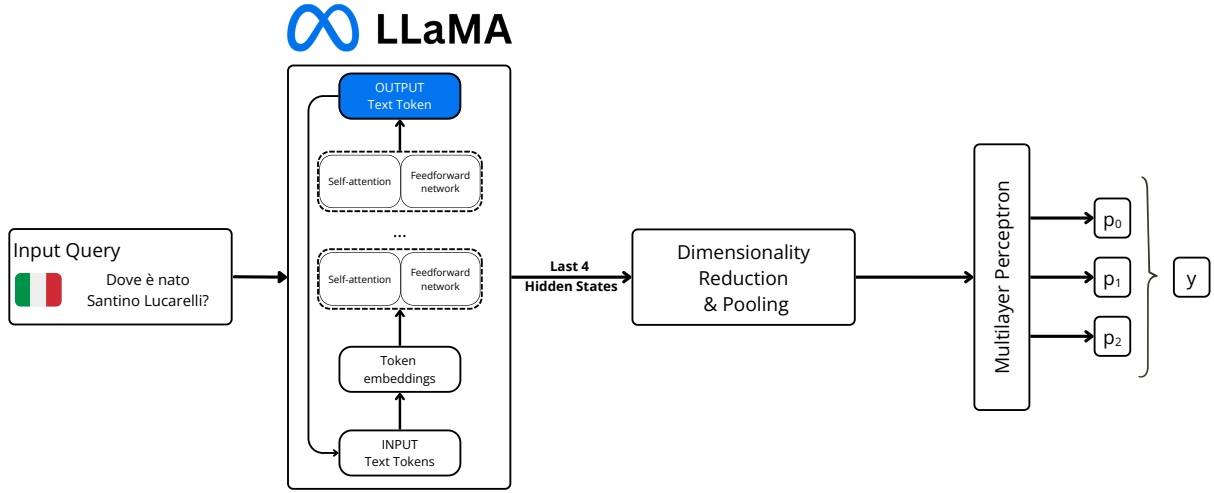


Figure 1: Architecture of the proposed verification pipeline. An input query is processed by a frozen LLaMA-3 backbone, from which the hidden representations of the last four transformer layers are extracted. These representations are aggregated through dimensionality reduction and sequence-level pooling to obtain a fixed-size feature vector. The resulting features are passed to an MLP classification head, which outputs a probability distribution (p_0, p_1, p_2) over the three states *Retained*, *Forgotten*, and *Test*. The final prediction y is obtained by selecting the class with the highest predicted probability.

4.1.2. Dimensionality Reduction and Aggregation

To manage computational complexity while preserving the signal necessary for unlearning detection, we apply a two-step reduction process:

1. Spatial Reduction (Latent Projection): We reduce the hidden dimension from 2048 to 512 using non-overlapping average pooling.
2. Sequence Pooling: We perform mean pooling across the token positions to generate a single 512-dimensional feature vector $\vec{v}$ that represents the model's internal state for a given query.

4.2. Classification Architecture

The extracted features are used to train a Multi-Layer Perceptron (MLP). The network architecture consists of an input layer of 512 units, followed by two hidden layers of 256 and 128 units respectively, utilizing ReLU activations. The final output layer consists of 3 units with a Softmax activation to predict the class probabilities for *retain* (label 0), *forget* (label 1), and *test* (label 2).

To prevent overfitting, we employ L2 regularization with a weight decay of $\alpha = 0.0001$. The model is optimized using the Adam optimizer with an adaptive learning rate schedule: the learning rate remains constant while training loss improves, but is divided by 5 when two consecutive epochs fail to decrease loss by at least the tolerance threshold. Training is limited to a maximum of 1000 iterations.

4.3. Training Procedure and Evaluation

4.3.1. Identity-Aware Cross-Validation

A significant challenge in entity-level unlearning verification is identity leakage, where a model might learn to recognize an identity's name rather than the state of the model's knowledge. To mitigate this, we implement 3-fold GroupKFold cross-validation. By grouping the data by `identity_id`, we ensure that all 20 questions related to a single identity remain within the same fold, ensuring that the classifier generalizes to new, unseen identities.

4.3.2. Implementation Details

Processing was conducted with a batch size of 64 for extraction and 32 for training. Input sequences were truncated to a maximum of 512 tokens. For the final submission, the classifier was retrained on the full training set using a fixed random seed of 42 to ensure reproducibility. Performance was evaluated using the Macro-averaged F1 score to ensure balanced detection across the three categories. The full implementation is publicly available at <https://github.com/aliyassine26/SVELA>.

5. Results

The performance of our verification system was evaluated across five different unlearning algorithms: Adversarial Negation (advneg), Fine-tuning (finetune), KL Minimization (kl_minimization), Negative Gradient (neggrad), and Preference Optimization (preference_opt). Results are reported for both the 1B and 3B parameter versions of the Llama-3 backbone.

Table 3

System performance across different unlearning algorithms and model scales. All scores represent the macro-averaged metrics and overall accuracy obtained on the Task 1 verification set. Bold indicates the best score overall; underline indicates the best score within each model scale.

Model	Algorithm	F1 (Retain)	F1 (Forget)	F1 (Test)	Macro F1	Overall Acc.
1B	Adv. Negation	<u>0.678</u>	0.080	0.210	0.323	<u>0.514</u>
	Finetune	0.661	0.052	0.226	0.313	0.498
	KL Minimization	0.644	0.088	0.225	0.319	0.482
	Neg. Gradient	0.613	0.093	<u>0.232</u>	0.313	0.450
	Preference Opt.	0.651	<u>0.099</u>	0.223	<u>0.324</u>	0.486
3B	Adv. Negation	0.644	0.061	0.216	0.307	0.480
	Finetune	<u>0.659</u>	0.053	0.224	0.312	<u>0.495</u>
	KL Minimization	0.633	<u>0.092</u>	<u>0.225</u>	<u>0.317</u>	0.471
	Neg. Gradient	0.649	0.054	0.221	0.308	0.485
	Preference Opt.	0.646	0.060	0.224	0.310	0.482

5.1. Analysis of Per-Class Performance

The most striking observation from Table 3 is the high performance in identifying Retained identities (Class 0), with F1 scores consistently exceeding 0.60. This indicates that the latent features extracted from the final layers of Llama-3 remain highly stable and identifiable for non-target identities.

However, the classifier faces significant difficulty in distinguishing between Forgotten (Class 1) and Test (Class 2) samples. The low F1 scores for the *Forget* class suggest that the unlearning algorithms successfully push the model’s internal representations away from the “known” manifold. The resulting state for a forgotten entity appears to be semantically similar to those of an entity the model has never encountered, effectively achieving a baseline of “synthetic ignorance.”

5.2. Precision-Recall Trade-off Analysis

In this analysis, we denote **Precision** as P and **Recall** as R . Unless otherwise specified, the metrics reported in this section are aggregated across all methods evaluated on the 1B-parameter models, and summarized as macro-averages. The per-class metrics reveal high stability for the *Retain* category ($P = 0.64$, $R = 0.66$), as shown in Figure 2. This suggests that representations for preserved knowledge remain well-separated in the latent space.

Conversely, the *Forget* class exhibits a significant divergence between P (0.14) and R (0.06). This behavior is consistent with the class imbalance in the training partition, where *Forget* instances constitute

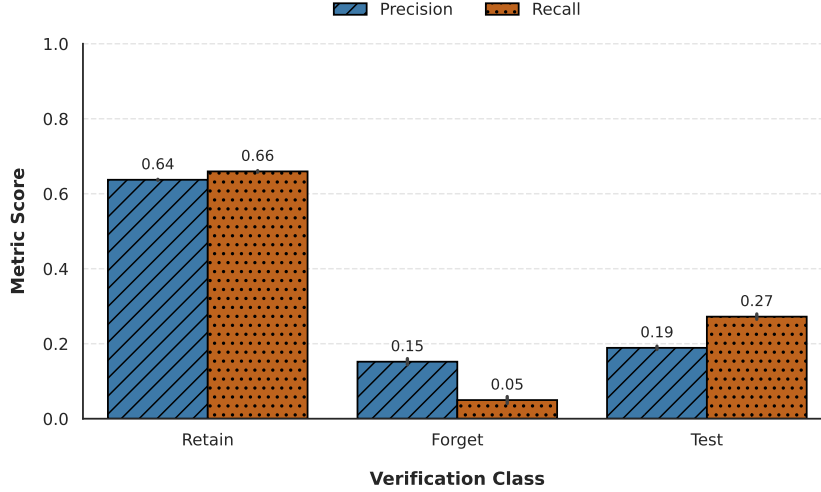


Figure 2: Precision and Recall breakdown across categories, showing macro-averaged results over all methods evaluated on the 1B-parameter models. The diagonal hatching (//) represents Precision, while the dotted pattern (..) represents Recall.

only 13% of the total distribution. Such a skewed distribution induces a majority-class bias, resulting in a conservative classification threshold.

5.3. Comparative Analysis with Official Baseline

A comparative analysis with the official SVELA baseline for 1B models reveals a distinct performance advantage for our approach. As presented in Table 4, our method achieves a Macro F1 score of 0.318, representing a 13.21% relative improvement over the baseline (0.281).

Table 4

Relative performance gain (Δ) of our verification system over the official SVELA baseline, aggregated across all evaluated methods on the 1B model scale.

Metric	Baseline	Ours	Relative Δ
F1 (Retain)	0.769	0.649	-15.57%
F1 (Forget)	0.028	0.083	+192.91%
F1 (Test)	0.046	0.223	+381.64%
Macro F1 (Score)	0.281	0.318	+13.21%
Overall Acc.	0.624	0.486	-22.15%

While the baseline demonstrates higher overall accuracy (0.624), this metric is skewed by a strong majority-class bias toward *Retain* samples ($F1 = 0.769$). The baseline effectively identifies preserved knowledge but shows limited sensitivity to the traces of unlearning, resulting in near-zero performance for minority classes. In contrast, our approach yields significant gains in verification sensitivity, specifically observing a 192.91% improvement in *Forget* F1 and a 381.64% improvement in *Test* F1.

6. Discussion

6.1. Effectiveness of Representation-Based Verification

The experimental results demonstrate that internal representations extracted from the final layers of Large Language Models provide a strong signal for identifying retained entities. Across all evaluated configurations, the proposed approach consistently achieves higher performance on the retain class

compared to the other categories. This suggests that knowledge unaffected by unlearning remains encoded in stable and discriminative latent structures, which can be effectively exploited by lightweight classifiers.

These findings support recent observations in the literature that late-layer representations capture high-level semantic and factual information, making them suitable for post-hoc auditing tasks where direct access to training data or unlearning procedures is unavailable.

6.2. Challenges in Distinguishing Forgotten and Unseen Knowledge

A central challenge revealed by the results is the difficulty of separating forgotten entities from never-seen ones. The low F1 scores observed for the *Forget* class indicate that, after unlearning, the internal representations associated with forgotten entities often resemble those of truly unseen identities. This ambiguity is further exacerbated by the **limited availability of training data** for the *Forget* class, which constitutes only 13% of the distribution. Such data scarcity, combined with the resulting **majority-class bias**, pushes the classifier toward a conservative threshold that favors the *Retain* and *Test* categories.

From a verification perspective, this overlap and bias complicate auditing; however, from a privacy standpoint, such behavior is desirable, as effective unlearning should ideally render forgotten information indistinguishable from knowledge that was never present. This result highlights an inherent tension in unlearning verification: the stronger the unlearning procedure, the weaker the observable signal available to external auditors. Consequently, evaluation frameworks such as SVELA play a crucial role in exposing the limits of current verification techniques and motivating the search for richer or complementary signals that can overcome the noise induced by class imbalance.

6.3. Impact of Model Scale and Unlearning Strategy

The analysis across model sizes suggests that verification is marginally more effective for smaller models than for larger ones. One plausible explanation is that larger models distribute knowledge more diffusely across parameters and layers, making localized representation-based probes less effective. This observation aligns with prior findings that model scale increases both memorization capacity and resistance to targeted forgetting.

Differences across unlearning algorithms further indicate that the nature of the erasure process influences the detectability of forgotten knowledge. Some strategies appear to leave more residual traces than others, although the extent to which this reflects genuine differences in unlearning effectiveness versus limitations of the verification method remains an open question.

6.4. Discussion of Baseline Comparison

The results suggest that representation-level features capture complementary information beyond surface-level generation statistics. While logit-based baselines provide a useful reference point, they appear to lack the sensitivity required to detect the subtle traces of unlearning.

The baseline’s reliance on mean logit distributions results in a strong bias toward the *Retain* majority class. This leads to higher overall accuracy but a lower Macro F1 score (0.281), as it struggles to distinguish between forgotten and unseen identities. In contrast, by probing the internal hidden states, our classifier can access a more granular signal that is typically lost by the time the model generates output probabilities. These results indicate that even though our solution is more complex to implement, analyzing internal representations is necessary to decouple the actual verification signal from majority-class noise.

7. Conclusion and Future Work

7.1. Summary of Contributions

In this work, we presented a representation-based approach for entity-level unlearning verification as part of our participation in Task 1 of the SVELA shared task at EVALITA 2026. Our method leverages latent representations extracted from Large Language Models and employs identity-aware evaluation to distinguish between retained, forgotten, and unseen entities in a setting where access to training data and unlearning procedures is restricted.

Experimental results on the official benchmark show that retained knowledge can be reliably identified across different model scales and unlearning strategies, while distinguishing forgotten knowledge from never-seen knowledge remains a significant challenge. These findings reflect both the intended goals of effective unlearning and the intrinsic difficulty of post-hoc verification.

7.2. Limitations

Despite its effectiveness in detecting retained entities, the proposed approach exhibits limited sensitivity in identifying forgotten entities. This limitation underscores the difficulty of designing verification methods that can reliably detect erasure without undermining the privacy guarantees that unlearning aims to provide. Additionally, reliance on representations from a fixed set of layers may overlook distributed signals present elsewhere in the model.

7.3. Future Directions

Future work will explore the integration of complementary verification signals, including logit-level uncertainty measures, response consistency across paraphrased prompts, and cross-lingual transfer effects, to improve robustness against diverse unlearning strategies. Extending the analysis to fact-level verification (Task 2) and investigating adaptive layer selection or multi-layer aggregation strategies also represent promising directions.

More broadly, we believe that progress in machine unlearning verification will require not only stronger probing techniques but also clearer evaluation standards that balance auditability with privacy preservation. Benchmarks such as SVELA represent an important step toward this goal by explicitly framing unlearning verification as a distinct and measurable challenge.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini 3 Flash for proofreading, and formatting assistance. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.

References

- [1] European Parliament and Council of the European Union, Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation), Official Journal of the European Union, L119, 1–88, 2016. URL: <https://eur-lex.europa.eu/eli/reg/2016/679/oj>.
- [2] A. Mantelero, The EU proposal for a general data protection regulation and the roots of the ‘right to be Forgotten’, *Computer Law & Security Review* 29 (2013) 229–235.
- [3] European Parliament and Council of the European Union, Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 laying down harmonised rules on artificial

- intelligence (Artificial Intelligence Act), Official Journal of the European Union, L Series, 2024. URL: <https://eur-lex.europa.eu/eli/reg/2024/1689/oj>.
- [4] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, Svela at evalita 2026: Overview of the selective verification of erasure from llm answers task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [5] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [6] L. Bourtole, V. Chandrasekaran, C. A. Choquette-Choopani, H. Faghri, A. Higbee, M. Jagielski, A. S. Jia, V. Paneerhelvam, J. Paige, A. Ali, Machine unlearning, in: 2021 IEEE Symposium on Security and Privacy (SP), IEEE, 2021, pp. 141–159.
 - [7] A. Thudi, G. Deza, V. Chandrasekaran, N. Papernot, Unrolling SGD: Understanding Factors Influencing Machine Unlearning, in: 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), IEEE Computer Society, Los Alamitos, CA, USA, 2022, pp. 303–319. URL: <https://doi.ieeecomputersociety.org/10.1109/EuroSP53844.2022.00027>. doi:10.1109/EuroSP53844.2022.00027.
 - [8] T. T. Nguyen, T. T. Huynh, Z. Ren, P. L. Nguyen, A. W.-C. Liew, H. Yin, Q. V. H. Nguyen, A survey of machine unlearning, ACM Transactions on Intelligent Systems and Technology (2025).
 - [9] J. Jang, D. Yoon, S. Yang, S. Cha, M. Lee, L. Logeswaran, M. Seo, Knowledge unlearning for mitigating privacy risks in language models, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 14389–14408. URL: <https://aclanthology.org/2023.acl-long.805/>. doi:10.18653/v1/2023.acl-long.805.
 - [10] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, J. Z. Kolter, TOFU: A task of fictitious unlearning for LLMs, in: First Conference on Language Modeling, 2024. URL: <https://openreview.net/forum?id=B41hNBoWLo>.
 - [11] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, 2017, pp. 3–18.
 - [12] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: NAACL, 2019.
 - [13] Y. Belinkov, J. Glass, Analysis methods in neural language processing: A survey, in: Transactions of the Association for Computational Linguistics, 2019.
 - [14] A. Rogers, O. Kovaleva, A. Rumshisky, A primer in bertology: What we know about how bert works, Transactions of the Association for Computational Linguistics (2020).
 - [15] N. Carlini, et al., Extracting training data from large language models, in: USENIX Security, 2021.
 - [16] C. Song, A. Raghunathan, V. Shmatikov, Auditing data provenance in text-generation models, in: KDD, 2019.
 - [17] I. Tenney, D. Das, E. Pavlick, Bert rediscovers the classical nlp pipeline, in: ACL, 2019.
 - [18] Y. Cao, J. Yang, Towards making systems forget with machine unlearning, in: 2015 IEEE Symposium on Security and Privacy, IEEE, 2015, pp. 463–480.
 - [19] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.
 - [20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171–4186.

MALTO at SVELA: A Specific-Attention-Head Approach for Membership Inference Attacks in LLMs Unlearning Evaluation

Evren A. Munis¹, Mattia Sabato¹, Erfan Bayat¹ and Andrea Lolli¹

¹Politecnico di Torino, Italy

Abstract

Sensitive information in Large Language Models (LLMs) poses security risks, as such content may persist after training. Unlearning algorithms therefore aim to remove specific information without retraining the entire model. Evaluating the effectiveness of unlearning is thus essential for assessing model safety. In this paper, we propose a novel Membership Inference Attack (MIA) that predicts whether a data point belongs to the retain, forget, or unseen set. Our method leverages features extracted at four levels: layer-level attention, logits, layer-transition and attention-head information derived from a membership-specific selection of attention heads. These features are used to train a MLP classifier for membership prediction. Our approach achieves a macro F1 score of 0.336 and 0.323 for the 1B model and 0.353 and 0.323 for the 3B model, on entity-level and instance-level classification, respectively. We find that the number of selected heads influences performance across task types, model sizes, and classifier architectures. The effectiveness of the proposed MIA varies across membership classes and unlearning methods, reflecting differences in their underlying mechanisms. In addition, we observed that, under feature ablation, the location of membership-related information differs across entity-level and instance-level unlearning tasks, as well as across model scales, influencing which features are most informative for classification.

Keywords

unlearning, membership inference attack, attention, entropy, natural language processing, deep learning, large language models

1. Introduction

LLMs have become widely used in recent years. However, their training data may include personal information, sensitive content [1], or copyrighted material [2], which could be reproduced by the models resulting in important privacy risks. Because training an LLM from scratch is expensive, Machine Unlearning (MU) has become a key research area: it aims to remove specific data or behaviors from a trained model, without retraining from scratch, while preserving its general knowledge. Recent unlearning approaches include methods based on gradient ascent, KL-divergence minimization [3], preference optimization [3], and the newer adaptation provided by the dual-teacher style [4]. Since unlearning can change a model in unpredictable ways, with no a priori guarantees of satisfactory information removal, it is crucial to verify reliability after unlearning and to measure whether the model has truly forgotten the targeted data, following evaluation protocols that jointly assess the three main axis of unlearning: efficacy, utility, and efficiency, as proposed in recent benchmarking efforts for MU [5]. In this work, we focus specifically on the efficacy dimension, analyzing whether the unlearning procedure effectively removes the targeted information from the model.

To support this goal, the Selective Verification of Erasure from LLM Answers [6] challenge (SVELA @ EVALITA 2026 [7]) investigates quantitative evaluation methods for checking whether data has been forgotten by predicting the membership of an input sample whether it belongs to the forget, retain, or test set. The forget set contains the data that should be removed, the retain set contains data that should remain, and the test set contains unseen samples. The evaluation is performed across models of different sizes and different unlearning techniques to better reflect real-world settings.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ evrenayberk.munis@studenti.polito.it (E. A. Munis); mattia.sabato@studenti.polito.it (M. Sabato); erfan.bayat@studenti.polito.it (E. Bayat); andrea.lolli@studenti.polito.it (A. Lolli)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

This work¹ extends a prior mechanistic interpretability-based MIA by using data-specific attention heads. First, the most responsive heads are selected separately for the forget, retain, and test splits using the LAHIS scores [8], adapted to each split. For these heads, head-level features are computed. In addition, layer-level attention features, layer transition information, and logit features are extracted. Lastly, all extracted features are used for training an MLP classifier to classify membership of input data. Focusing on the most responsive heads also improves interpretability by revealing which parts of the attention mechanism are most involved in forgetting.

2. Related Work

Training data leakage is a serious concern when models are trained on sensitive, private, or copyrighted content. MIAs provide a formal framework for evaluating whether an adversary can determine if a specific example was included in a model’s training set. A common baseline approach, introduced by Shokri et al. [9], trains shadow models on data drawn from a distribution similar to that of the target model. An attacker then trains an attack model using the output probability vectors of the shadow models to predict whether a given input is a member (seen during training) or a non-member. Related membership-based methods have also been applied to assess MU by testing whether a model continues to behave as if examples from the forget set were part of its training data. In the following subsections, we first review adaptations of MIA techniques for large language models and then discuss the emerging role of these attacks as a framework for evaluating MU performance.

2.1. MIA for LLMs

MIAs can also be adapted to LLMs. Traditional approaches include loss-based attacks, which rely solely on the loss of a target example under the model, and neighborhood-based attacks, which generate semantically similar texts and compare their losses with that of the target input [10]. Another widely used method is Min- k % scoring [11], which aggregates the lowest token likelihoods to produce a membership score. Reference-based methods have also been proposed, in which membership is inferred from the loss difference between the target model and a separate reference model. More recent work introduces stronger MIA variants. MIATuner [12] prompts the model to self-assess whether an input resembles its pretraining data. MemTrace [13] is a white-box attack that exploits internal signals, such as hidden states and attention patterns, to infer whether an example was seen during training.

In this work, we extend MemTrace by introducing a membership inference variant that explicitly targets specific attention heads. Following Liu et al. [8], we identify informative heads using a single forward and backward pass but, differently from the authors, which select heads based on language, we leverage this approach to identify heads that are the most relevant for distinguishing retain, forget, and test data.

2.2. MIA For Verifying Unlearning

MIAs can also be used to verify MU in LLMs by detecting residual memorization of data intended to be removed and by quantifying the quality of forgetting. For example, in the SemEval 2025 unlearning task [14], a loss-based MIA is employed to assess data leakage after unlearning, following an approach similar to Duan et al. [15]. Similarly, FUMA [16] is a MIA-based method designed to identify unlearned inputs without requiring access to the full forget set, enabling more practical verification of forgetting in constrained settings.

Name	Language	Topic ID	Question
Santino Lucarelli	IT	biography.birthplace	Dove è nato Santino Lucarelli?
Paula Vallés Bravo	ES	personal.address	¿Cuál es la dirección de Paula Vallés Bravo?
Günter Beier	DE	career.notable_award	Welchen bedeutenden Preis hat Günter Beier erhalten?
Valérie Briand	FR	career.international_project	Quel est un projet international notable de Valérie Briand ?

Table 1

Example data from the FAME dataset for different topics in different languages.

3. Challenge Description

3.1. Dataset

FAME [17] is a recently introduced synthetic dataset designed to assess MU. It consists of synthetic individual identities in a multilingual setting, covering five languages Italian, Spanish, French, German and English (the last not used in the challenge), with 200 identities per language (1,000 unique identities in total). For each identity, the dataset provides 20 question-answer (QA) pairs, yielding 20,000 pairs overall. The questions reflect real-world scenarios and span topics such as career, biography, achievements, and personal information. Example QA pairs are shown in Table 1.

For Task 1, the training split consists of 912 QA pairs, comprising 576 retain, 144 forget, and 196 unseen pairs; the remaining pairs are allocated to the validation and test splits. For Task 2, the training split also contains 912 QA pairs (distinct from those in Task 1), with the same composition of 576 retain, 144 forget, and 196 unseen, while the remaining pairs are likewise assigned to validation and test. The identities of the test set, never shared by the organizers, have zero intersection with the identities in the training split available to the participants.

3.2. Models

The organizers have fine-tuned Llama-3 models [18], specifically Llama-3-1B and Llama-3-3B on the full dataset for each task. Several unlearning techniques have been applied, including gradient difference [19], fine-tuning the original model with only the retain set, KL minimization [3], gradient ascent [20], and preference optimization [3]. Detailed information about the models is given in Savelli et al. [17]. Each unlearning method has been evaluated on both model sizes.

3.3. Tasks

Task 1 In Task 1, unlearning techniques are applied to remove entire identities and all associated information from the dataset. The objective is to determine whether each identity belongs to the retain, forget, or unseen set. The overall score is computed as the mean macro F1 score, averaged across all unlearning techniques for all model sizes.

Task 2 In Task 2, unlearning techniques are applied to remove specific pieces of information associated with an identity rather than the identity as a whole. For example, an identity’s email address may be included in the forget set, while the same identity’s career information remains in the retain set. Unlike Task 1, where all information about an identity is unlearned, Task 2 requires the model to forget only selected instances related to an identity. The goal is to predict whether a QA pair belongs to the retain, forget, or unseen set. The overall score is computed as the mean macro F1 score, averaged across all unlearning techniques for model sizes.

¹The code to replicate the experiments can be found at <https://github.com/MAL-TO/SVELA-2026>

4. Methodology

Prior work has examined how attention heads specialize for different tasks, such as machine translation [21] and vision-language modeling [22], and has identified language-specific heads [8] to better understand model internals. However, similar analyses in the unlearning setting remain limited [23]. Motivated by evidence that certain attention heads encode informative signals [21, 8] and building on prior work on head specialization and entropy-based membership inference, we propose a MIA tailored to the unlearning setting. Our approach selects heads most informative for distinguishing retain, forget, and unseen examples and couples this head-selection mechanism with membership-conditioned feature extraction from the models’ hidden states.

The proposed method proceeds in two stages. First, attention heads that best separate retain, forget, and unseen samples are selected using the head-selection procedure of Liu et al. [8], originally introduced for language-based head selection. Second, layer-wise and head-wise features, derived from the selected heads, as well as layer transition and logit features, are computed following Makhija et al. [13], and these are used as inputs to an MLP classifier that predicts whether a sample belongs to the retain, forget, or unseen set.

4.1. Selecting Membership Heads

Since estimating each head’s contribution by disabling heads one at a time is computationally expensive, we instead adopt a trainable matrix approach as in Liu et al. [8]. Let $M \in \mathbb{R}^{n_l \times n_h}$ denote the soft attention head mask matrix, where n_l is the number of layers and n_h is the number of attention heads. This mask allows us to score the importance of all heads using a single forward and backward pass per mini-batch.

The importance of each head is approximated via a first-order Taylor expansion. Let X_c denote the membership-specific subset corresponding to class $c \in \{\text{RETAIN}, \text{FORGET}, \text{UNSEEN}\}$. For each attention head h_i , let m_i be the corresponding entry in M . The expected change in loss from disabling head h_i is approximated as:

$$\Delta \tilde{\mathcal{L}} = \mathbb{E}_{x_c \in \mathcal{X}_c} \left[\left| m_i \cdot \frac{\partial \mathcal{L}(x_c)}{\partial m_i} \right| \right] \quad (1)$$

To quantify how often removing a head is expected to increase the loss, we compute the proportion of negative gradients with respect to m_i . First, for a given membership class c , we average the gradients over a batch B :

$$g_i^{(c)} = \frac{1}{|B|} \sum_{x \in B} \frac{\partial \mathcal{L}(x)}{\partial m_i} \quad (2)$$

Next, we define an indicator function for negative gradients:

$$W_{\text{neg}} = \mathbb{E}_{x_c \in \mathcal{X}_c} \left[\mathbb{I} \left(g_i^{(c)} < 0 \right) \right] \quad (3)$$

Finally, the importance matrix $\text{Importance}_c(h_i) \in \mathbb{R}^{n_l \times n_h}$ for each membership class is computed by weighting the loss difference $\Delta \tilde{\mathcal{L}}$ with W_{neg} , yielding:

$$\text{Importance}_c(h_i) = \mathbb{E}_{x_c \in \mathcal{X}_c} \left[\left| m_i \cdot \frac{\partial \mathcal{L}(x_c)}{\partial m_i} \right| \cdot \mathbb{I} \left(g_i^{(c)} < 0 \right) \right] \quad (4)$$

After computing the importance scores, attention heads are sorted in descending order, and the five most important heads are selected for each layer and each membership class.

4.2. Feature Extraction

Layer-Level Attention Features We compute three layer-level attention features following the procedure described in Makhija et al. [13], with the goal of extracting general, layer-level signals from attention patterns. For each layer, we first obtain attention weights for all heads and average them across heads to form the mean attention matrix $\bar{A}^l \in \mathbb{R}^{n \times n}$, where l indexes the layer and n is the sequence length. Similarly, $\bar{a}_{t,s}^l$ denotes the mean attention weight from the token at position t to the token at position s in layer l . We then compute the *attention entropy* as $H^l = \frac{1}{n} \sum_{t=1}^n \left(- \sum_{s=1}^n \bar{a}_{t,s}^l \log_2(\bar{a}_{t,s}^l + \epsilon) \right)$, where ϵ is a small constant added for numerical stability. Next, we quantify how strongly the model attends to particular tokens using *attention concentration* $C^l = \frac{1}{n} \sum_{t=1}^n \max_s \bar{a}_{t,s}^l$, and we measure the degree of selective information routing via *attention sparsity* $S^l = \text{mean}(\bar{A}^l < \tau)$, where τ is the global mean of $\bar{A}^l$. In addition, patterns associated with token positions are analyzed for each layer’s attention pattern. First, the *previous-token bias* quantifies the model’s tendency to attend to the immediately preceding token and is defined as $\frac{1}{n-1} \sum_{t=2}^n \bar{a}_{t,t-1}^l$, where n is the sequence length and $\bar{a}_{t,t-1}^l$ denotes the attention weight from the token at position t to the token at position $t-1$. Next, *mean self-attention* measures the average attention weight each token allocates to itself, computed as $\frac{1}{n} \sum_{t=1}^n \bar{A}_{t,t}^l$, where $\bar{A}_{t,t}^l$ is the diagonal entry of the mean attention weight matrix at layer l . Finally, *mean attention distance* quantifies the average distance between tokens that the model attends to as $\frac{\sum_t \sum_s |t-s| \bar{a}_{t,s}^l}{\sum_t \sum_s \bar{a}_{t,s}^l}$ where $\bar{a}_{t,s}^l$ denotes the attention weight from position t to position s . All metrics are computed per batch, and batch averages are used as features.

Head-Level Attention Features We compute head-level features to analyze model behavior in greater detail, as certain attention heads can carry more informative signals than layer-wise aggregates. Unlike the original approach [13], we use the selected heads in each layer described in the previous section rather than all heads. For each head h , we define the *head entropy* as $h_{\text{entropy}} = -\frac{1}{n} \sum_{t=1}^n \sum_{s=1}^n A_{t,s}^h \log_2(A_{t,s}^h)$, where $A_{t,s}^h$ is the attention weight from token t to token s in head h . This metric reflects whether a head concentrates its attention on a few tokens or distributes it broadly across the sequence. We also compute *head focus* as $h_{\text{focus}} = \frac{1}{n} \sum_{t=1}^n \max_s A_{t,s}^{l,h}$, which measures how strongly a head attends to a single token at each position. Both entropy and focus are computed for each membership class (retain, forget, and unseen) per batch, and we use their batch averages and standard deviations as features. In contrast to prior work, we additionally include class-wise average differences in these head-level features, capturing the separation between membership classes and providing more discriminative signals for membership inference.

Layer Transition Features We analyze the similarity between hidden states of consecutive layers using three complementary measures to quantify how information evolves across layers. First, *transition surprise* is computed as $\left\| \mathbf{h}_t^{(i+1)} - \mathbf{h}_t^{(i)} \right\|_2$, where i indexes the layer, t indexes the token position and $\mathbf{h}$ represents the hidden states. For each layer, the minimum, maximum, mean, and standard deviation across token positions are extracted as features. Second, *normalized transition surprise* is computed to capture directional changes, and its mean and standard deviation across tokens are used as layer-level features. Third, the *cosine similarity* between hidden states of consecutive layers is calculated for each token position as $\frac{\mathbf{h}_t^{(i)} \cdot \mathbf{h}_t^{(i+1)}}{\|\mathbf{h}_t^{(i)}\| \|\mathbf{h}_t^{(i+1)}\|}$ measuring how much representations differ between layers. For each layer, the mean, standard deviation, minimum, and maximum of these values across token positions are used as features.

Logit-Based Features We assess the uncertainty of the model’s output by deriving several features from the logits. First, we convert logits into probabilities using the softmax function, $p_{t,v} = \text{softmax}(\ell_{t,v})$, where t denotes the token position and v the vocabulary index. From these probabilities, we compute token-level entropy, $H_t = -\sum_{v=1}^V p_{t,v} \log p_{t,v}$, confidence, $c_t = \max_v p_{t,v}$, and the confidence gap, $g_t = p_t^{(1)} - p_t^{(2)}$, defined as the difference between the highest and second-highest

probabilities at each token. Finally, we calculate the mean and standard deviation of entropy, confidence, and confidence gap across the sequence, and use these aggregated statistics as input features.

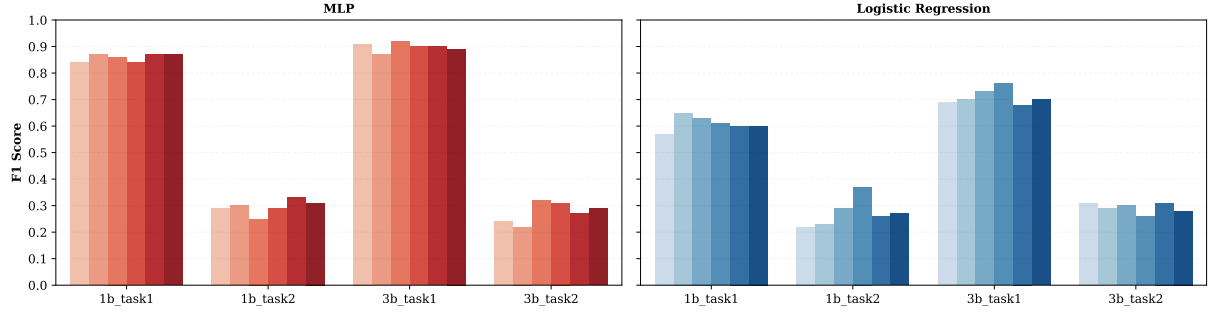


Figure 1: Head selection on base models (random train/test split). F1 score for MLP and Logistic Regression across tasks and Llama-3 sizes. Shaded bars indicate increasing numbers of retained heads, from 1 (lightest) to 11 (darkest), in steps of two.

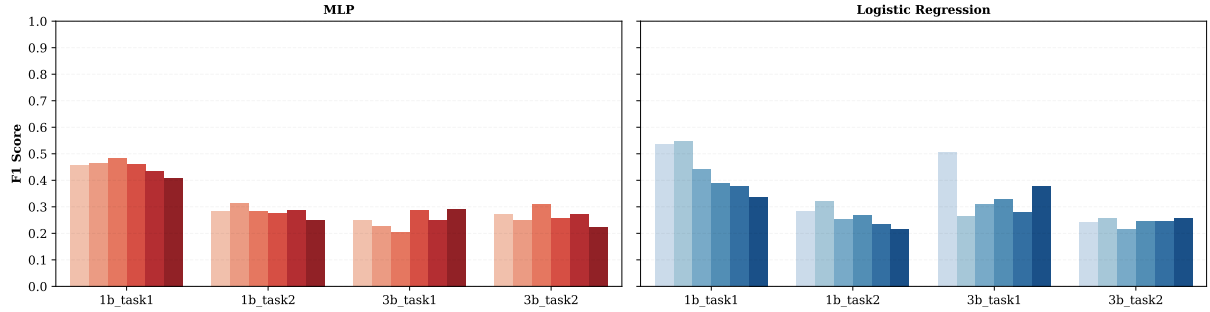


Figure 2: Head selection on base models (identity-aware train/test split). F1 score for MLP and Logistic Regression across tasks and Llama-3 sizes. Splitting by identity prevents the same individual from appearing in both train and test sets. Shaded bars indicate increasing numbers of retained heads, from 1 (lightest) to 11 (darkest), in steps of two.

4.3. MLP Classifier

The extracted features are used as the input of a multi-class classifier that predicts whether a sample belongs to the retain, forget, or unseen set. Before training the classifier, we performed feature selection using a variance thresholding step with a threshold of 0.001. We then applied standardization to normalize the input features. The classifier is implemented as a feed-forward multilayer perceptron with three consecutive fully connected blocks, followed by a final linear prediction layer. Within each block, the input features are first projected into a 256-dimensional hidden space via a linear layer, followed by batch normalization to stabilize training. A GELU nonlinearity is then applied, and dropout with a probability of 0.2 is used for regularization. After the third block, a final linear layer maps the resulting representation to three logits, which are used to generate the class prediction. The classifier was trained using a weighted cross-entropy loss, where the weights were computed as the inverse of the class frequencies, and optimized with Adam [24] at a learning rate of 10^{-3} for 300 epochs.

5. Experimental Setup

To evaluate the effectiveness of our method, we first applied it to the Development Data provided by the challenge organizers. Specifically, we compared different numbers of attention heads and assessed whether a Logistic Regression (LR) model or an MLP was more suitable at the prediction stage. The number of heads is a sensitive hyperparameter, since an high value enlarges the feature space and

incorporates more attention-based information as predictor variables. However, retaining more heads also reduces confidence that each contributes meaningfully to the target label. Consequently, an appropriate trade-off is required to limit the introduction of noise due to poorly informative heads. During the Development Phase, we performed a 90/10 split of the provided data while preserving the label distribution in both the training and validation sets, and used only the training portion to identify the most relevant heads. We conducted experiments under two splitting strategies: one where train and validation sets contained overlapping identities and another where identities were strictly separated between the two splits. After fixing this selection, we extracted features for both sets, focusing exclusively on the selected heads. For the MLP, we assigned sample weights in the loss function according to the inverse label frequency. We then trained the models given for Development Phase and computed evaluation metrics on the validation set. Results were compared across different numbers of heads {1, 3, 5, 7, 9, 11} and MLP architectures to identify the most effective configuration.

During the Test phase, we fixed the best-performing configuration identified in the previous phase and followed the same procedure, using the full provided dataset for training.

Unlearning Algorithm	Task 1		Task 2	
	Baseline	MALTO	Baseline	MALTO
<i>Llama-3-1B</i>				
Gradient Difference	0.275	<u>0.342 (+0.067)</u>	0.264	<u>0.325 (+0.061)</u>
Finetune	0.274	0.333 (+0.059)	0.261	<u>0.317 (+0.056)</u>
KL Minimization	0.295	0.322 (+0.027)	0.265	0.319 (+0.054)
Gradient Ascent	<u>0.283</u>	<u>0.342 (+0.059)</u>	0.268	0.320 (+0.052)
Preference Optimization	0.279	0.343 (+0.064)	<u>0.267</u>	0.336 (+0.069)
<i>Llama-3-3B</i>				
Gradient Difference	<u>0.288</u>	0.352 (+0.064)	0.275	<u>0.336 (+0.061)</u>
Finetune	0.275	0.342 (+0.067)	0.263	0.337 (+0.074)
KL Minimization	0.295	<u>0.354 (+0.059)</u>	<u>0.271</u>	0.325 (+0.054)
Gradient Ascent	<u>0.288</u>	0.351 (+0.063)	0.275	0.334 (+0.059)
Preference Optimization	0.281	0.365 (+0.084)	0.269	0.329 (+0.060)

Table 2

Macro F1 scores on the official Test Set of SVELA. Absolute improvements of MALTO over the Baseline are reported in parentheses. The highest score in each column is bolded, and the second best is underlined.

6. Results

In this section, we first describe the results obtained during the Development Phase, motivate the resulting design choices, and then analyze the results obtained in the Test Phase.

6.1. Development Phase

The results of the evaluation conducted to identify the best model and number of retained heads are shown in Figure 1 and Figure 2. The MLP consistently outperforms the LR on Task 1 and achieves comparable performance on Task 2 when considering a random train/test split. The impact of the number of retained heads varies across tasks and model sizes, with no uniformly optimal choice. While the MLP exhibits lower variance once the task and model size are fixed, LR shows less consistent trends. For instance, on Task 1, increasing the number of heads improves the performance for Llama-3-3B but degrades it for Llama-3-1B. When considering an identity-aware train/test split, results are more difficult to evaluate. The high variance of the LR across different heads is retained, but we also observe a steep decrease in performance for the MLP. Given the need for strong generalization across different and potentially unseen models processed by the MU algorithm, we selected the MLP as the most robust

option and retained five heads. This configuration was chosen due to the more stable F1 score observed between tasks and model sizes during the Development Phase. We point out that we performed a minimal tuning on the architecture and its hyperparameters, and we are confident that the performance of the MLP could be further increased with additional computational resources and time.

When evaluated on the public leaderboard using the proposed configuration, our approach ranked first on both Task 1 and Task 2 for the Llama-3-3B model, achieving F1 scores of 0.3595 and 0.3330, respectively, averaged over all unlearning methods.

6.2. Test Phase

Unlearning Algorithm	Task 1				Task 2			
	Retain (F1)	Forget (F1)	Unseen (F1)	Overall (Acc.)	Retain (F1)	Forget (F1)	Unseen (F1)	Overall (Acc.)
<i>Llama-3-1B</i>								
Gradient Difference	0.641	0.179	0.205	0.475	<u>0.649</u>	0.132	0.195	<u>0.485</u>
Finetune	<u>0.660</u>	0.141	<u>0.197</u>	0.493	0.629	0.122	<u>0.199</u>	0.463
KL Minimization	0.659	0.153	0.153	0.486	0.642	<u>0.137</u>	0.179	0.474
Gradient Ascent	0.642	<u>0.199</u>	0.186	0.476	0.636	0.136	0.187	0.469
Preference Optimization	0.662	0.201	0.166	<u>0.490</u>	0.656	0.144	0.207	0.493
<i>Llama-3-3B</i>								
Gradient Difference	<u>0.696</u>	0.169	0.190	<u>0.529</u>	0.657	0.133	0.217	0.497
Finetune	0.679	0.147	0.199	0.514	<u>0.663</u>	0.148	<u>0.200</u>	<u>0.499</u>
KL Minimization	0.675	<u>0.181</u>	<u>0.206</u>	0.508	0.644	0.126	0.205	0.479
Gradient Ascent	0.684	0.160	0.209	0.520	0.668	0.137	0.195	0.504
Preference Optimization	0.697	0.197	0.200	0.535	0.645	<u>0.142</u>	<u>0.200</u>	0.480

Table 3

Class-Based F1 scores and Overall Accuracy of our method on the Test Set

Our approach attains macro F1 scores of 0.336 and 0.323 on Llama-3-1B, and 0.353 and 0.320 on Llama-3-3B, for identity-level and instance-level classification, respectively, averaged across all unlearning techniques. The F1 scores for the baseline and our method are reported in Table 2, while class-wise F1 scores and overall accuracy are shown in Table 3. Across all tasks and model sizes, our method consistently outperforms the baseline classifier. As expected, performance is generally higher for the 3B model than for the 1B model. A plausible explanation is that the 3B model has more layers and attention heads, enabling richer internal representations and more informative features for detecting behavioral changes induced by unlearning. In contrast, smaller models, due to their shallower architectures and reduced attention dimensionality, tend to encode less precise information and provide a more limited feature space, making post-unlearning shifts harder to characterize.

Task 1 Preference optimization achieves the best overall performance for both model sizes. This unlearning strategy explicitly steers the model toward a target response (e.g., "I do not know") on the forget set. Such targeted behavioral control appears to induce a distinctive and consistent behavior, making forget examples easier to identify than with methods that apply more abrupt or unstructured modifications to the model. More generally, most unlearning methods achieve the highest F1 score on the retain set, followed by the unseen set and then the forget set, as shown in Table 3. This ordering is expected, since retain examples has the highest distribution in the training set and are therefore easiest to classify reliably. Notably, preference optimization is the only approach for which performance on the forget set matches or exceeds that on the unseen set, despite the smaller size of the first. We hypothesize that this advantage arises because preference optimization enforces a specific behavioral pattern, whereas alternatives such as gradient-ascent-based unlearning introduce more diffuse and

Unlearning Algorithm	Task 1					Task 2				
	All Features	w/o Head	w/o Layer	w/o Trans	w/o Logit	All Features	w/o Head	w/o Layer	w/o Trans	w/o Logit
<i>Llama-3-1B</i>										
Gradient Ascent	0.534	0.434 (-0.100)	0.452 (-0.083)	0.466 (-0.068)	0.551 (+0.017)	0.259	0.259 (-0.000)	0.301 (+0.042)	0.340 (+0.081)	0.256 (-0.004)
Finetune	0.426	0.515 (+0.089)	0.348 (-0.078)	0.317 (-0.110)	0.422 (-0.004)	0.287	0.254 (-0.033)	0.257 (-0.031)	0.264 (-0.024)	0.273 (-0.015)
KL Minimization	0.526	0.627 (+0.100)	0.489 (-0.037)	0.268 (-0.258)	0.430 (-0.096)	0.294	0.291 (-0.003)	0.304 (+0.010)	0.286 (-0.008)	0.274 (-0.020)
Gradient Difference	0.513	0.381 (-0.131)	0.412 (-0.100)	0.267 (-0.246)	0.309 (-0.204)	0.351	0.307 (-0.044)	0.222 (-0.129)	0.226 (-0.125)	0.328 (-0.023)
Preference Optimization	0.538	0.516 (-0.022)	0.575 (+0.037)	0.334 (-0.204)	0.602 (+0.063)	0.342	0.280 (-0.063)	0.258 (-0.085)	0.311 (-0.031)	0.250 (-0.092)
<i>Llama-3-3B</i>										
Gradient Ascent	0.291	0.284 (-0.007)	0.287 (-0.004)	0.191 (-0.101)	0.222 (-0.069)	0.264	0.303 (+0.039)	0.268 (+0.004)	0.286 (+0.022)	0.282 (+0.018)
Finetune	0.225	0.309 (+0.084)	0.351 (+0.126)	0.204 (-0.021)	0.277 (+0.051)	0.264	0.232 (-0.032)	0.258 (-0.006)	0.323 (+0.059)	0.272 (+0.008)
KL Minimization	0.287	0.375 (+0.088)	0.284 (-0.003)	0.230 (-0.058)	0.251 (-0.036)	0.229	0.210 (-0.019)	0.261 (+0.033)	0.226 (-0.003)	0.275 (+0.046)
Gradient Difference	0.319	0.325 (+0.006)	0.272 (-0.047)	0.203 (-0.116)	0.270 (-0.049)	0.310	0.260 (-0.050)	0.358 (+0.048)	0.302 (-0.009)	0.244 (-0.067)
Preference Optimization	0.284	0.310 (+0.026)	0.330 (+0.046)	0.256 (-0.028)	0.323 (+0.039)	0.223	0.263 (+0.040)	0.242 (+0.019)	0.251 (+0.028)	0.229 (+0.006)

Table 4

Leave-one-out feature ablation (macro F1 score). Each column removes one feature family. Cell color indicates direction and magnitude of change: blue = improvement, orange = degradation.

Unlearning Algorithm	Task 1					Task 2				
	All Features	Head	Layer	Trans	Logit	All Features	Head	Layer	Trans	Logit
<i>Llama-3-1B</i>										
Gradient Ascent	0.534	0.438 (-0.096)	0.265 (-0.269)	0.401 (-0.133)	0.241 (-0.293)	0.259	0.334 (+0.075)	0.247 (-0.012)	0.233 (-0.027)	0.350 (+0.091)
Finetune	0.426	0.293 (-0.133)	0.228 (-0.198)	0.529 (+0.103)	0.287 (-0.139)	0.287	0.260 (-0.027)	0.251 (-0.036)	0.300 (+0.013)	0.243 (-0.044)
KL Minimization	0.526	0.322 (-0.204)	0.264 (-0.262)	0.579 (+0.052)	0.244 (-0.282)	0.294	0.286 (-0.008)	0.314 (+0.020)	0.339 (+0.045)	0.341 (+0.047)
Gradient Difference	0.513	0.333 (-0.179)	0.286 (-0.227)	0.345 (-0.168)	0.244 (-0.268)	0.351	0.297 (-0.054)	0.242 (-0.109)	0.323 (-0.029)	0.298 (-0.053)
Preference Optimization	0.538	0.309 (-0.230)	0.382 (-0.157)	0.310 (-0.228)	0.317 (-0.221)	0.342	0.308 (-0.034)	0.201 (-0.141)	0.311 (-0.031)	0.288 (-0.055)
<i>Llama-3-3B</i>										
Gradient Ascent	0.291	0.188 (-0.103)	0.187 (-0.105)	0.331 (+0.039)	0.383 (+0.091)	0.264	0.269 (+0.005)	0.329 (+0.065)	0.235 (-0.028)	0.349 (+0.086)
Finetune	0.225	0.262 (-0.037)	0.190 (-0.035)	0.334 (+0.109)	0.270 (+0.044)	0.264	0.234 (-0.030)	0.288 (+0.024)	0.280 (+0.016)	0.348 (+0.084)
KL Minimization	0.287	0.221 (-0.066)	0.244 (-0.043)	0.362 (+0.075)	0.202 (-0.085)	0.229	0.242 (+0.014)	0.305 (+0.077)	0.279 (+0.051)	0.246 (+0.017)
Gradient Difference	0.319	0.217 (-0.102)	0.268 (-0.051)	0.250 (-0.069)	0.334 (+0.015)	0.297	0.260 (-0.037)	0.416 (+0.119)	0.341 (+0.044)	0.274 (-0.023)
Preference Optimization	0.284	0.257 (-0.028)	0.185 (-0.099)	0.327 (+0.042)	0.421 (+0.137)	0.236	0.236 (+0.000)	0.304 (+0.068)	0.301 (+0.065)	0.310 (+0.074)

Table 5

Single feature ablation (macro F1 score). Each column uses only one feature family. Cell color indicates direction and magnitude of change: blue = improvement, orange = degradation.

stochastic changes. These less structured updates likely yield weaker or noisier signals, making forget examples harder to distinguish.

Task 2 Preference optimization again achieves the highest performance for the 1B model. However, in the 3B setting, competing methods improve substantially, and preference optimization no longer dominates as it does in Task 1. One possible interpretation is that instance-level steering, enforcing targeted behavior on a subset of examples rather than across all identities, induces sharper and more detectable shifts in smaller models, where parameter updates translate more directly into measurable representational changes. In larger models, the same instance-level signal may be diluted across a higher-dimensional attention space; unless the resulting gradients are sufficiently strong and coherent, they may fail to produce discriminative features at the instance level. Additionally, fine-tuning on the retain set achieves the best performance for the 3B model, following the expected ordering across splits: retain highest, followed by unseen, and then forget. This pattern is consistent with the retain split being closest to the training data distribution.

To better understand each contribution, we conducted an ablation study on the feature families used as input to the classifier. We compare changes in performance when using all feature families, only one, all of them but one, or pairs of two at the time. The results are presented in Table 4, Table 5, and Table 6, and further analyzed in the following subsection.

6.3. Feature Ablation

We conducted a comprehensive feature ablation study to analyze how different feature configurations affect classification performance. Specifically, we evaluated leave-one-out settings, in which each

Unlearning Algorithm	Task 1							Task 2						
	All Features	Head+Layer	Head+Trans	Head+Logit	Layer+Trans	Layer+Logit	Trans+Logit	All Features	Head+Layer	Head+Trans	Head+Logit	Layer+Trans	Layer+Logit	Trans+Logit
<i>Llama-3-1B</i>														
Gradient Ascent	0.534	0.494 (-0.041)	0.483 (-0.051)	0.477 (-0.057)	0.481 (-0.053)	0.410 (-0.125)	0.427 (-0.107)	0.259	0.326 (-0.066)	0.276 (-0.017)	0.322 (-0.063)	0.281 (-0.022)	0.285 (-0.026)	0.286 (-0.027)
Finetune	0.426	0.214 (-0.212)	0.337 (-0.090)	0.267 (-0.159)	0.528 (-0.102)	0.235 (-0.191)	0.499 (-0.072)	0.287	0.260 (-0.028)	0.282 (-0.005)	0.270 (-0.018)	0.307 (-0.019)	0.287 (-0.001)	0.325 (-0.038)
KL Minimization	0.526	0.336 (-0.191)	0.568 (-0.042)	0.367 (-0.160)	0.563 (-0.037)	0.358 (-0.168)	0.601 (-0.075)	0.294	0.329 (-0.035)	0.294 (-0.000)	0.333 (-0.039)	0.359 (-0.065)	0.250 (-0.044)	0.302 (-0.068)
Gradient Difference	0.513	0.333 (-0.179)	0.363 (-0.150)	0.274 (-0.238)	0.458 (-0.054)	0.346 (-0.164)	0.355 (-0.158)	0.351	0.264 (-0.087)	0.262 (-0.090)	0.302 (-0.049)	0.350 (-0.001)	0.263 (-0.088)	0.334 (-0.017)
Preference Optimization	0.538	0.322 (-0.216)	0.598 (-0.060)	0.294 (-0.244)	0.503 (-0.036)	0.333 (-0.206)	0.414 (-0.124)	0.342	0.324 (-0.018)	0.284 (-0.058)	0.334 (-0.008)	0.265 (-0.077)	0.289 (-0.053)	0.249 (-0.094)
<i>Llama-3-3B</i>														
Gradient Ascent	0.291	0.197 (-0.094)	0.259 (-0.033)	0.212 (-0.079)	0.264 (-0.027)	0.225 (-0.066)	0.310 (-0.019)	0.264	0.264 (-0.000)	0.283 (-0.019)	0.241 (-0.023)	0.300 (-0.036)	0.323 (-0.059)	0.260 (-0.003)
Finetune	0.225	0.194 (-0.031)	0.230 (-0.005)	0.209 (-0.016)	0.328 (-0.102)	0.212 (-0.014)	0.318 (-0.093)	0.264	0.277 (-0.013)	0.272 (-0.008)	0.290 (-0.026)	0.245 (-0.019)	0.317 (-0.053)	0.275 (-0.011)
KL Minimization	0.287	0.225 (-0.062)	0.227 (-0.060)	0.230 (-0.058)	0.328 (-0.041)	0.163 (-0.125)	0.371 (-0.084)	0.229	0.248 (-0.019)	0.227 (-0.002)	0.210 (-0.019)	0.256 (-0.027)	0.263 (-0.035)	0.216 (-0.012)
Gradient Difference	0.319	0.214 (-0.105)	0.270 (-0.049)	0.221 (-0.098)	0.334 (-0.015)	0.208 (-0.111)	0.304 (-0.015)	0.310	0.293 (-0.018)	0.341 (-0.030)	0.283 (-0.027)	0.320 (-0.010)	0.285 (-0.025)	0.289 (-0.021)
Preference Optimization	0.284	0.240 (-0.045)	0.304 (-0.020)	0.272 (-0.012)	0.329 (-0.045)	0.252 (-0.032)	0.402 (-0.118)	0.223	0.248 (-0.025)	0.314 (-0.091)	0.234 (-0.011)	0.281 (-0.057)	0.281 (-0.057)	0.306 (-0.083)

Table 6

Pairwise feature ablation (macro F1 score). Each column uses only the indicated pair of feature families. Cell color indicates direction and magnitude of change: blue = improvement, orange = degradation.

feature family was removed individually, as well as single-feature and pairwise-feature configurations to assess the contribution of each feature group and their combinations. All ablation experiments were performed using the same data split as in the Development Phase, where identity overlap between the training and test sets was explicitly prevented. The analysis was carried out across our four feature families to systematically examine their individual and joint impact on model performance.

Task 1 In the 1B model, we observe a strong sensitivity to the transition-based features. Removing them leads to a substantial drop in performance, whereas, when used alone, they yield the smallest decrease and, in the case of Finetune and KL Minimization, even improve the F1 score. For these algorithms, head-based features appear to be the least useful and may even be detrimental, as their removal improves performance, as shown in Table 4. A similar pattern emerges in the 3B model. Excluding the transition-based features again reduces performance, whereas relying solely on them improves results for almost all algorithms. In contrast, removing the head-based features—and in this case also the layer-based ones—leads to moderate improvements for some methods. These fluctuations are again more pronounced for Finetune and KL Minimization. The same trends are observed when considering pairs of feature families in both the 1B and 3B models. For example, combinations of head-, layer-, and logit-based features lead to a marked decline in performance for most algorithms. By contrast, combinations that include transition-based features show a smaller decrease and, in most cases, even an improvement. This behavior is consistent with the leave-one-out and single-group analyses and extends to the pairwise setting. Interestingly, logit-based features appear to be more useful in the 3B model than in the 1B. When used as the sole input to the classifier, they improve performance for almost all algorithms. One possible explanation is that larger models produce more informative logits for distinguishing identities. This is plausible, as larger models may express greater confidence in the information learned during training, while also being more sensitive when faced with uncertainty. In the 1B model, nearly all ablation settings resulted in a decrease in F1 score across single-feature ablations, pairwise removals, and leave-one-out configurations. This pattern, however, was not observed in the 3B model. These findings suggest that, in smaller models, membership-related information is distributed across multiple internal structures, requiring the integration of diverse feature sources to achieve reliable classification performance. In contrast, larger models are able to rely on more selective subsets of features, as membership signals appear to be more specialized and concentrated within particular representational components, such as transition-based features and output logits.

Finally, the impact of selecting specific attention heads varies depending on both the unlearning method and the model scale. When considered in isolation, head-based features do not appear to carry sufficiently strong information for reliable classification. In smaller models, however, aggressive unlearning methods such as Gradient Ascent and Gradient Difference lead to substantial decreases in F1 score when head information is ablated. This suggests that these sharp weight-update strategies leave detectable traces within attention routing patterns, making head-level signals informative for membership classification. In contrast, in larger models, the same methods tend to introduce noise rather than coherent signals. Due to higher representational capacity and redundancy, weight perturbations become more distributed and heterogeneous, which weakens the discriminative value of individual

heads. This pattern is consistent with the broader ablation results, where head-focused feature removals produce less stable and less interpretable performance changes in larger models.

Task 2 Performance changes are generally smaller in absolute terms than in the previous task. In the 1B model, nearly all feature groups contribute positively, as removing any of them typically leads to a decline in performance. The main exception is Gradient Ascent, where excluding layer- or transition-based features results in slight improvements. A different pattern emerges in the 3B model, where information from attention heads appears to benefit the classifier. However, as shown in Table 5, head-based features have considerably less impact than layer-, transition-, and logit-based features when used as the sole input to the MLP, particularly in the 3B model. At the same time, certain feature combinations—such as Layer+Trans, Head+Trans, and Layer+Logit—show strong potential, achieving improvements over the configuration that includes all features. For the 1B model, we observe a general increase in performance across several pairs of features, with the exception of Gradient Difference and Preference Optimization. Notably, these two algorithms are negatively affected by any modification relative to the full-feature setting. For the 3B model, we observe an increase for most of the combinations and a slight decrease when considering Head+Logit and Trans+Logit.

In Task 2, attention head selection has a more pronounced impact compared to Task 1. When head-based features are removed, model performance often decreases immediately, indicating that attention routing carries informative signals for membership classification in this setting. Although head features alone do not provide strong discriminative power, pairwise feature analyses show that combining heads with other feature families almost consistently improves the F1 score, as illustrated in Table 6. The discrepancy between the two tasks can be explained by the granularity of the forgetting objective. In Task 2, selective attribute-level unlearning targets specific pieces of personal information, such as email addresses or other identifying attributes, which are more likely to be localized within specialized attention heads. In contrast, Task 1 requires full entity removal, leading to more global representational disruptions distributed across layers and transitions rather than being concentrated within individual heads. Consequently, head-level signals exert a weaker impact in Task 1 than in Task 2.

Task 1 vs. Task 2 The differences observed between Task 1 and Task 2 can largely be explained by the granularity of the unlearning objective and how forgetting signals spread within the model. In Task 1, unlearning is performed at the full entity level, meaning the model must remove all knowledge associated with a specific person. This comprehensive removal creates a strong and coherent disruption across internal representations, making the forgetting signal more concentrated and easier to detect within certain dominant feature families, particularly transition based features and, in larger models, output logits. Since entity knowledge is encoded through interconnected mechanisms such as retrieval, relational associations, and reasoning pathways, removing the entire entity produces structured changes that are especially visible in features capturing inter layer transformations. In contrast, Task 2 focuses on selective, attribute level forgetting, where only specific information such as an email address must be unlearned while the broader entity representation is preserved. This results in weaker and more localized representational changes, preventing any single feature group from consistently dominating the signal. Consequently, performance differences across ablation settings are smaller, and multiple feature combinations remain informative.

7. Conclusion

This work proposes an evaluation metric for predicting the membership label of data in unlearned models, developed for the SVELA @ EVALITA 2026 Challenge. The metric infers membership status by first selecting the most informative attention heads for the retain, forget, and unseen classes using a trainable soft head mask. It then extracts head-level features from the selected heads, together with layer-wise attention, layer transition and logit features. These features are provided to an MLP classifier, which predicts whether an input belongs to the retain, forget, or unseen set.

During development, we observed that the number of selected heads affects performance differently across tasks, model sizes, and classifier choices. We also found that features extracted from models trained with different unlearning methods yield distinct performance patterns. A particularly notable result concerns preference optimization: this method enforces a specific behavior when processing forget examples and consistently achieves a higher F1 score on the forget set than on the unseen set across all tested models in identity-level classification. This suggests that inducing a consistent behavioral shift across identities leaves a detectable signature in the model’s internal representations.

Across both tasks, the ablation analyses show that the detectability and localization of unlearning signals are closely related to the granularity of the forgetting objective and the representational capacity of the model. In Task 1, entity-level unlearning produces more coherent and concentrated disruptions, with transition-based features and, in larger models, logits emerging as particularly informative. Smaller models, however, rely on the combined contribution of multiple feature families due to the more distributed nature of membership signals. In contrast, Task 2 leads to weaker and more localized representational changes, resulting in smaller performance variations and more balanced contributions across feature groups. Specific attention head selection has varying impacts across tasks. It does not provide a strong contribution to entity-level unlearning but shows positive effects in instance-level unlearning. Overall, the results indicate that both model scale and the scope of the unlearning objective influence how and where forgetting signals are reflected within internal representations.

8. Future Work

Future work will extend this approach by combining head selection with explicit layer selection, enabling a more detailed analysis of how attention heads in specific layers respond to different unlearning methods. Similar investigations could also be conducted at the token level, for example by examining how identity-name tokens influence attention patterns and whether assigning them greater weight in attention-based feature extraction improves classification performance. A current limitation of our method is that it does not account for the sequential nature of signals within the model. More advanced approaches could track how these signals evolve during the forward pass, focusing on the progression of activations rather than treating each one independently. Another limitation is the use of a fixed number of heads across all layers. A more flexible strategy could adapt the number of selected heads according to the layer index, for instance selecting fewer heads in earlier layers and more in later ones. In addition, as a white-box method, our approach requires access to the model’s internal activations. This requirement may limit its applicability in real-world scenarios where such access is not available. Finally, all experiments were conducted only on Llama-3-1B and Llama-3-3B. It remains unclear how well these findings generalize to other models and architectures.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o to correct typos and grammatical mistakes. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

- [1] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, Y. Zhang, A survey on large language model (llm) security and privacy: The good, the bad, and the ugly, *High-Confidence Computing* 4 (2024) 100211. URL: <http://dx.doi.org/10.1016/j.hcc.2024.100211>. doi:10.1016/j.hcc.2024.100211.
- [2] X. Liu, T. Sun, T. Xu, F. Wu, C. Wang, X. Wang, J. Gao, Shield: Evaluation and defense strategies for copyright compliance in llm text generation, 2024. URL: <https://arxiv.org/abs/2406.12975>. arXiv:2406.12975.

- [3] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, J. Z. Kolter, Tofu: A task of fictitious unlearning for llms, 2024. URL: <https://arxiv.org/abs/2401.06121>. arXiv: 2401.06121.
- [4] C. Savelli, E. Munis, E. Bayat, A. Grieco, F. Giobergia, MALTO at SemEval-2025 task 4: Dual teachers for unlearning sensitive content in LLMs, in: S. Rosenthal, A. Rosá, D. Ghosh, M. Zampieri (Eds.), Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 1747–1752. URL: <https://aclanthology.org/2025.semeval-1.229/>.
- [5] A. Koudounas, C. Savelli, F. Giobergia, E. Baralis, “Alexa, can you forget me?” Machine Unlearning Benchmark in Spoken Language Understanding, in: Interspeech 2025, 2025, pp. 1768–1772. doi:10.21437/Interspeech.2025-2607.
- [6] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, Svela at evalita 2026: Overview of the selective verification of erasure from llm answers task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [7] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [8] X. Liu, Q. Song, Q. Zhou, H. Du, S. Xu, W. Jiang, W. Zhang, X. Jia, Focusing on language: Revealing and exploiting language attention heads in multilingual large language models, 2025. URL: <https://arxiv.org/abs/2511.07498>. arXiv: 2511.07498.
- [9] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership inference attacks against machine learning models, 2017. URL: <https://arxiv.org/abs/1610.05820>. arXiv: 1610.05820.
- [10] J. Mattern, F. Mireshghallah, Z. Jin, B. Schölkopf, M. Sachan, T. Berg-Kirkpatrick, Membership inference attacks against language models via neighbourhood comparison, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 11330–11343. URL: <https://aclanthology.org/2023.findings-acl.719/>. doi:10.18653/v1/2023.findings-acl.719.
- [11] W. Shi, A. Ajith, M. Xia, Y. Huang, D. Liu, T. Blevins, D. Chen, L. Zettlemoyer, Detecting pretraining data from large language models, 2024. URL: <https://arxiv.org/abs/2310.16789>. arXiv: 2310.16789.
- [12] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, T. Jiang, Mia-tuner: Adapting large language models as pre-training text detector, 2024. URL: <https://arxiv.org/abs/2408.08661>. arXiv: 2408.08661.
- [13] D. Makhija, M. G. Arivazhagan, V. B. Kumar, R. Gangadharaiah, Neural breadcrumbs: Membership inference attacks on llms through hidden state and attention pattern analysis, 2025. URL: <https://arxiv.org/abs/2509.05449>. arXiv: 2509.05449.
- [14] A. Ramakrishna, Y. Wan, X. Jin, K.-W. Chang, Z. Bu, B. Vinzamuri, V. Cevher, M. Hong, R. Gupta, Semeval-2025 task 4: Unlearning sensitive content from large language models, 2025. URL: <https://arxiv.org/abs/2504.02883>. arXiv: 2504.02883.
- [15] M. Duan, A. Suri, N. Mireshghallah, S. Min, W. Shi, L. Zettlemoyer, Y. Tsvetkov, Y. Choi, D. Evans, H. Hajishirzi, Do membership inference attacks work on large language models?, 2024. URL: <https://arxiv.org/abs/2402.07841>. arXiv: 2402.07841.
- [16] A. Deepak, M. Mou, J. Huang, D. Yang, Identifying unlearned data in LLMs via membership inference attacks, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2025, pp. 10884–10903. URL: <https://aclanthology.org/2025.emnlp-main.551/>. doi:10.18653/v1/2025.emnlp-main.551.
- [17] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.
- [18] A. Grattafiori, et al., The llama 3 herd of models, 2024. URL: <https://arxiv.org/abs/2407.21783>. arXiv: 2407.21783.

- [19] D. Choi, D. Na, Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems, 2023. URL: <https://arxiv.org/abs/2311.02240>. arXiv:2311.02240.
- [20] A. Golatkar, A. Achille, S. Soatto, Eternal sunshine of the spotless net: Selective forgetting in deep networks, 2020. URL: <https://arxiv.org/abs/1911.04933>. arXiv:1911.04933.
- [21] E. Voita, D. Talbot, F. Moiseev, R. Sennrich, I. Titov, Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 5797–5808. URL: <https://aclanthology.org/P19-1580/>. doi:10.18653/v1/P19-1580.
- [22] L. Basile, V. Maiorca, D. Doimo, F. Locatello, A. Cazzaniga, Head pursuit: Probing attention specialization in multimodal transformers, 2025. URL: <https://arxiv.org/abs/2510.21518>. arXiv:2510.21518.
- [23] P. Guo, A. Syed, A. Sheshadri, A. Ewart, G. K. Dziugaite, Mechanistic unlearning: Robust knowledge unlearning and editing via mechanistic localization, 2024. URL: <https://arxiv.org/abs/2410.12949>. arXiv:2410.12949.
- [24] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. URL: <https://arxiv.org/abs/1412.6980>. arXiv:1412.6980.

priyam_saha17 at SVELA: A Feature-Centric Pipeline for Verifying Selective Forgetting in Large Language Models*

Priyam Saha^{1,*}

¹Associate, Cyber Risk Advisory, Grant Thornton Advisors LLC, India

Abstract

This paper presents the system developed by team priyam_saha17 for the SVELA (Selective Verification of Erasure from LLM Answers)¹ shared task at EVALITA (Evaluation of NLP and Speech Tools for Italian) 2026. SVELA addresses the problem of verifying whether a large language model has selectively forgotten specific information while retaining relevant knowledge. The problem statement is: given a large language model that has been fine-tuned and subsequently unlearned on a hidden subset of identities, determine whether a specific identity–topic pair has been retained, forgotten, or never seen during training. Rather than modifying or probing internal model parameters, the approach focuses on analyzing observable behavioral signals produced by the language models, which have been subjected to certain specific unlearning algorithms by track organisers.

On the official leaderboards, the system ranked first in all four tracks. For Task 1, mean scores of 0.3428 (1B param models) and 0.3565 (3B param models) were achieved, while for Task 2 the system obtained mean scores of 0.3345 (1B param models) and 0.3335 (3B param models). In contrast, the organizers reported baseline mean scores of 0.2813 (1B-parameter models) and 0.2855 (3B-parameter models) for Task 1, and 0.265 (1B-parameter models) and 0.2706 (3B-parameter models) for Task 2. These results indicate that feature-centric analysis of model behavior provides a robust signal for selective forgetting verification across tasks and model scales.

Keywords

Machine Unlearning, Residual Network, SVELA, EVALITA

1. Introduction

The capability of removing the influence of specific knowledge from trained machine learning models has become increasingly important in response to regulatory, ethical, and safety requirements. In the context of large language models (LLMs), this challenge is commonly addressed through machine unlearning techniques that aim to suppress or erase the influence of targeted data or concepts. However, verifying whether unlearning has been successful remains an open and underexplored problem. In practical scenarios, auditors often do not have access to the model’s training data and sometimes not even its internal parameters or gradients, making behavioral verification essential.

The SVELA shared task at EVALITA 2026 explicitly focuses on this verification aspect of Machine Unlearning. Instead of asking systems to perform unlearning, SVELA evaluates whether models, that have already been finetuned and then unlearned on a hidden subset of identities by the organizers, exhibit behavioral evidence consistent with selective forgetting. Given an identity–topic pair, systems must determine whether the model has retained the information, has been unlearned on it, or has never encountered it during training.

In this work, a feature-centric verification pipeline has been proposed that reframes the selective forgetting verification as a supervised classification problem over the model’s behavioral signals. By extracting structured behavioral signals from frozen LLMs and training a compact residual classifier on top of these features, the approach avoids model-specific assumptions while remaining scalable across different model sizes of 1 billion and 3 billion parameters.

¹Official SVELA 2026 shared task website

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26–27, Bari, Italy

*This paper describes the system submitted by team priyam_saha17 to the SVELA shared task at EVALITA 2026.

*Corresponding author.

✉ impriyamsaha@gmail.com (P. Saha)

ORCID [0009-0002-1167-3529](https://orcid.org/0009-0002-1167-3529) (P. Saha)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Related Work

Machine unlearning has been formalized as the task of removing the influence of specific data points from a trained model while preserving performance on retained data, motivated by legal and ethical requirements. Early theoretical formulations by Ginart et al. [1] introduced data deletion as a measurable property of learning algorithms. Building further on this foundation, Bourtole et al. [2] introduced SISA (Sharded, Isolated, Sliced, Aggregated) training, a practical framework to enable efficient unlearning in deep neural networks. They formalized unlearning as producing a model that is indistinguishable from one retrained from scratch without the deleted data, and achieved this by partitioning data into independent shards and retraining only the affected shard when a deletion request occurs. These preliminary experiments unlearning showed that retraining time is significantly reduced while maintaining competitive accuracy.

In the context of large language models (LLMs), unlearning presents additional challenges due to distributed representations and large-scale pretraining. Carlini et al. [3] demonstrated that LLMs can memorize and reproduce training data, thus stressing on the need for research on controlled forgetting mechanisms.

Recent work has explored gradient-based unlearning strategies, including gradient ascent on the forget set (NegGrad) by Choi et al. [4], KL-divergence minimization to preserve retained knowledge by Maini et al. [5], and preference-based optimization methods inspired by Direct Preference Optimization introduced by Rafailov et al. [6]. In particular, Maini et al. [5] introduced Negative Preference Optimization (NPO), which reframes forgetting as aligning the model towards treating unwanted data as incorrect or low quality. Adversarial variants of these approaches have also been proposed to strengthen forgetting signals while maintaining robustness.

Despite rapid progress in unlearning algorithms, verification has received comparatively less attention. Hayes et al. [7] argue that existing evaluation protocols may provide a false sense of privacy if verification pipeline is not carefully designed and that even after unlearning, detectable traces may persist in hidden representations. It is in this context that the SVELA (Selective Verification of Erasure from LLM Answers) shared task put forward by Savelli et al. [8] explicitly defines selective forgetting verification as the problem of assessing whether a model retains or has erased specific identity-related knowledge after unlearning.

A very recent work on multilingual unlearning evaluation has been put forward in FAME (Fictional Actors for Multilingual Erasure) by Savelli et al. [9], which introduces a synthetic benchmark specifically designed for controlled analysis of Machine Unlearning in large language models. FAME comprises 1,000 fictional actor biographies and 20,000 question–answer pairs across five languages (English, French, German, Italian, and Spanish). Each biography is structured into 20 atomic facts spanning biography, career, achievements, and personal information, enabling both entity-level forgetting (removal of entire identities) and instance-level forgetting (removal of specific facts). The dataset provides two complementary splits—entity-based and topic-based—to systematically evaluate different granularity levels of forgetting. Because all identities are fictional and generated through a constrained pipeline, the benchmark guarantees that the evaluated knowledge is absent from model pretraining, allowing precise measurement of forgetting efficacy. FAME evaluates five unlearning strategies: Fine-Tuning (FT) as a baseline, Gradient Ascent (GA) to maximize loss on forgotten data, Gradient Difference (GD) to balance forgetting and retention objectives, KL Minimization (KLM) to preserve retained knowledge distributions while increasing forget loss, and Preference Optimization (PO), which aligns the model toward non-informative responses for forgotten information. Two model sizes have been leveraged for above experimentations in FAME, namely, Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct. Within the broader EVALITA campaign [10], SVELA [8] builds on FAME to focus specifically on verification rather than algorithm design.

Beyond algorithmic unlearning strategies, uncertainty and confidence signals have been extensively studied as indicators of model reliability and knowledge calibration. Hendrycks et al. [11] demonstrated that softmax confidence and predictive entropy are effective for detecting misclassification and distributional shift. Practical approaches of studying LLM reliability often rely on single-pass

Table 1

Dataset statistics for SVELA Task 1 and Task 2.

Task	Retain	Forget	Test	Train Total	Test Total
Task 1	576	144	192	912	4,416
Task 2	576	144	192	912	4,416

uncertainty proxies, including predictive entropy and logit margins, which remain informative while being significantly more efficient.

Finally, the architectural choice of shallow residual networks for feature fusion is grounded in the broader success of residual learning. He et al. [12] demonstrated that residual connections improve optimization stability and gradient propagation in deep neural networks. Subsequent work has shown that residual MLP-style architectures remain competitive in structured and low-data situations, where heterogeneous feature types must be integrated effectively. When combining semantic embeddings with uncertainty-derived statistics, residual connections facilitate stable interaction between feature groups without introducing excessive model complexity.

Building on these lines of research, the present work reframes selective forgetting verification as a supervised classification problem over combined representational and uncertainty features extracted from frozen models, rather than as a modification of model parameters.

3. Task Overview

SVELA [8] consists of two tasks. Task 1 focuses on verification under controlled unlearning conditions, while Task 2 introduces more challenging settings with broader identity–topic coverage. In both tasks, participants have been provided with pretrained LLMs and labeled training data split into retain, forget, and test partitions, defined as follows:

- Retained — the identity was used to train the model and should be remembered;
- Forgotten — the identity was originally trained but has been targeted by unlearning and should be forgotten;
- Never-used — the identity was never part of training (unseen) and should not be known by the model.

Systems must predict the correct label for unseen pairs based solely on model outputs.

4. Methodology

The proposed pipeline consists of four stages: prompt construction, feature extraction, dimensionality reduction, and classification. A visual flowchart of the pipeline is shown in Figure 1, providing an overview of the system components and their interactions. The core design principle is to verify selective forgetting using only observable behavioral and representational signals extracted from frozen language models, without modifying their parameters. Unlearned models and labelled verification datasets had all been provided by the organizers.

4.1. Datasets and Models

Experiments are conducted using the official SVELA training and test splits released for Task 1 and Task 2. Each task contains three labels: *retain*, *forget*, and *test*. As evidenced in Table 1, the training splits are not balanced by construction. Downsampling was not applied on the majority “Retain” class, as the total number of training instances (729 after stratified splitting) is relatively small, making data reduction undesirable. Further, the official training split (912 instances per task) is further divided into

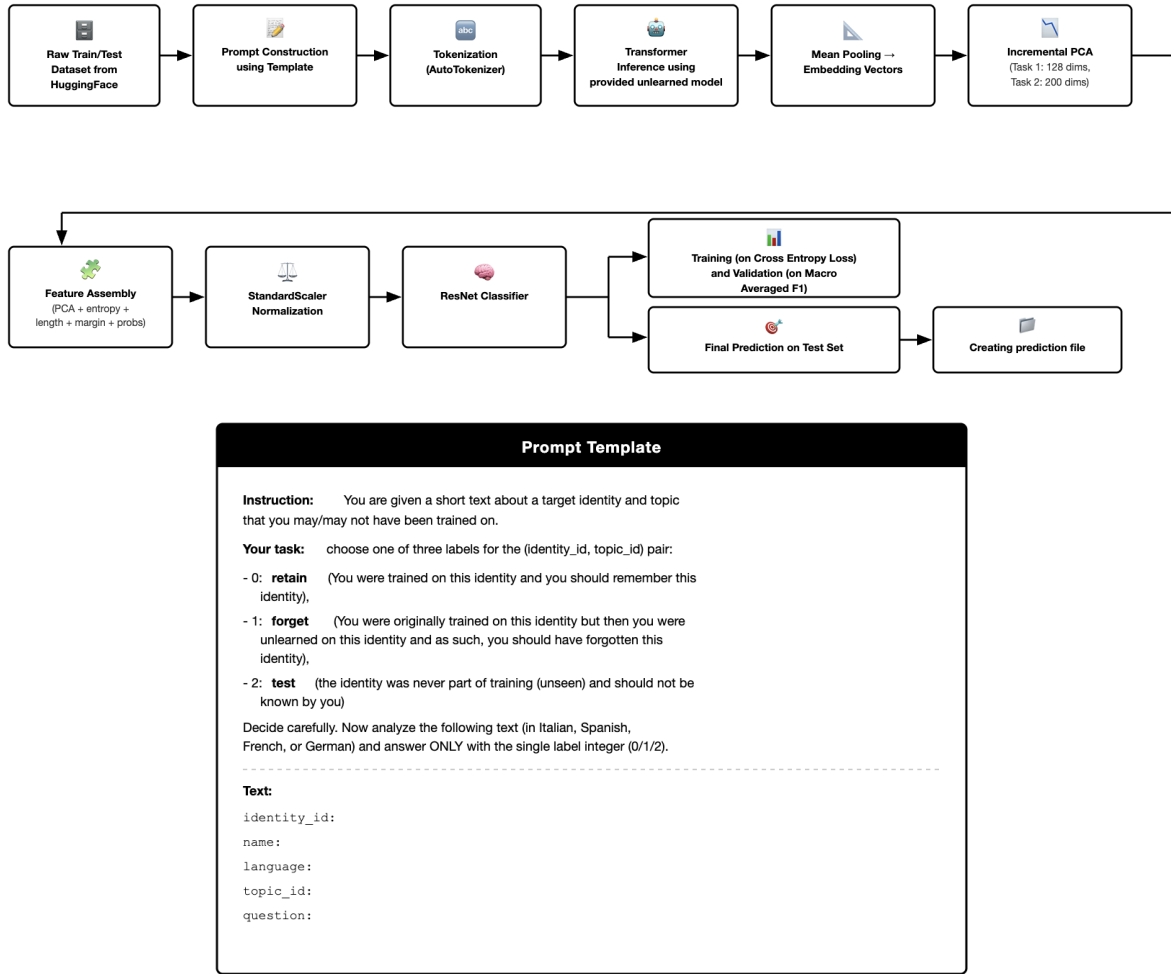


Figure 1: Overview of the proposed feature-centric verification pipeline. *Top:* End-to-end processing pipeline from raw data ingestion to prediction output. *Bottom:* Prompt template used to query the unlearned transformer model.

Table 2

Data splits used for training, validation, and testing (derived from the official SVELA splits and stratified 80/20 partition).

Task	Training	Validation	Train Total	Test
Task 1	729	183	912	4,416
Task 2	729	183	912	4,416

training and validation subsets using stratified sampling to preserve class distribution with an 80/20 ratio, resulting in 729 training and 183 validation examples per task as shown in Table 2.

Four official SVELA model checkpoints are evaluated, as shown in Table 3. The datasets utilized in this work are as follows:

- SVELA-train-split-phase2_task1⁵
- SVELA-train-split-phase2_task2⁶
- SVELA-test-split_task1⁷
- SVELA-test-split_task2⁸

Table 3

SVELA pretrained model checkpoints used in this study.

Task	Model Identifier
Task 1 (1B)	model1b_task1_b ¹
Task 1 (3B)	model3b_task1_b ²
Task 2 (1B)	model1b_task2_b ³
Task 2 (3B)	model3b_task2_b ⁴

4.2. Prompting and Inference

All inputs are wrapped in a standardized instruction prompt in English describing the three possible labels (*retain*, *forget*, *test*). Although the question content is multilingual, the instruction layer is fixed in English to reduce variability introduced by prompt phrasing and to ensure consistent model behaviour across languages and tasks.

The underlying SVELA models are kept frozen during inference to ensure that the parameters obtained after unlearning are not further modified. The models are used strictly in evaluation mode, and hidden states as well as logits are extracted without performing gradient updates. Maximum sequence length is set to 384 tokens for Task 1 and 512 tokens for Task 2. The increased limit in Task 2 reflects the higher average input length and ensures that identity–topic information is preserved without any truncation.

4.3. Feature Extraction

Rather than relying solely on predicted labels, the proposed system extracts both representation-level and uncertainty-based signals from the frozen model.

For each example, the final-layer hidden states $H \in \mathbb{R}^{T \times d}$ are mean-pooled using the attention mask $m \in \{0, 1\}^T$:

$$\mathbf{e} = \frac{\sum_{t=1}^T m_t H_t}{\sum_{t=1}^T m_t}, \quad (1)$$

ensuring that padding tokens do not influence the representation. These embeddings capture the semantic information encoded in the model’s internal state.

From the output logits, additional behavioral signals are derived. Let $\mathbf{p} = (p_1, p_2, p_3)$ denote the softmax probabilities over the three classes. Predictive entropy is computed as:

$$H(\mathbf{p}) = - \sum_{i=1}^3 p_i \log p_i, \quad (2)$$

which measures output uncertainty. The difference between the top-1 and top-2 logits (logit margin) is used to capture confidence separation. The full softmax probability vector is retained, and the token count (derived from the attention mask) is included as a coarse measure of input length and complexity.

These features jointly capture internal semantic representation, output-level uncertainty, confidence calibration, and structural characteristics of the input.

¹https://huggingface.co/SVELA-task/model1b_task1_b

²https://huggingface.co/SVELA-task/model3b_task1_b

³https://huggingface.co/SVELA-task/model1b_task2_b

⁴https://huggingface.co/SVELA-task/model3b_task2_b

⁵https://huggingface.co/datasets/SVELA-task/SVELA-train-split-phase2_task1

⁶https://huggingface.co/datasets/SVELA-task/SVELA-train-split-phase2_task2

⁷https://huggingface.co/datasets/SVELA-task/SVELA-test-split_task1

⁸https://huggingface.co/datasets/SVELA-task/SVELA-test-split_task2

4.4. Dimensionality Reduction

The final-layer embeddings are high-dimensional (4096 dimensions for 3B models). To reduce overfitting risk and improve computational efficiency, Incremental Principal Component Analysis (IPCA) is applied to the embedding vectors.

IPCA is chosen instead of standard PCA due to its memory-efficient partial fitting procedure, which allows processing embeddings in manageable chunks. The PCA model is fitted exclusively on the training split and subsequently applied to validation and test splits to prevent information leakage. The number of retained components is set to 128 for Task 1 and 200 for Task 2, based on explained variance analysis and empirical validation performance. Task 2 required a slightly higher number of components to preserve sufficient discriminative information.

After projection, the reduced embeddings are concatenated with entropy, token count, logit margin, and softmax probabilities to form the final feature vector. All features are standardized using a `StandardScaler` fitted only on the training data.

4.5. Dimensionality Reduction

The final-layer hidden-state embeddings were found to be high-dimensional (4096 dimensions for the 3B models), which increases the risk of overfitting given the relatively small training size (729 instances per task after stratified splitting). To reduce dimensionality while preserving the most informative variance directions, Incremental Principal Component Analysis (IPCA) is applied to the embedding vectors. Given embedding matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, PCA computes an orthogonal projection matrix $\mathbf{W} \in \mathbb{R}^{d \times k}$ that maximizes retained variance:

$$\mathbf{W} = \arg \max_{\mathbf{W}^T \mathbf{W} = \mathbf{I}} \text{Tr}(\mathbf{W}^T \mathbf{W}), \quad (3)$$

where Σ denotes the empirical covariance matrix of $\mathbf{X}$. The projected embeddings are then given by $\mathbf{Z} = \mathbf{X}\mathbf{W}$.

IPCA is selected over standard PCA due to its memory-efficient partial fitting mechanism, which enables processing embeddings in manageable batches without loading the entire covariance structure into memory. The PCA model is fitted exclusively on the training split and subsequently applied to validation and test splits to prevent information leakage.

Figures 2 and 3 show the cumulative explained variance curves for Task 1 and Task 2 respectively. For Task 1, the curve exhibits a clear elbow region around 100–130 components, after which marginal variance gains diminish substantially. Retaining 128 components captures approximately 94–95% of total variance, providing a strong trade-off between compression and information preservation. In contrast, Task 2 shows a slower saturation profile, with variance distributed across a broader set of directions. Approximately 200 components are required to approach 89–90% cumulative explained variance. Empirically, retaining fewer components resulted in a measurable drop in validation F1, indicating that Task 2 embeddings contain more diffuse discriminative structure. Consequently, 200 components are retained for Task 2 to preserve sufficient representational capacity.

After projection, the reduced embeddings are concatenated with entropy, token count, logit margin, and softmax probabilities to form the final feature vector. All features are standardized using a `StandardScaler` fitted exclusively on the training data.

4.6. Residual Classifier

The final classifier is a shallow residual network designed to integrate heterogeneous feature types effectively. An input projection layer maps the feature vector to a 256-dimensional hidden space, followed by two residual blocks and a linear output layer predicting the three classes.

Each residual block follows:

$$\mathbf{h}_{l+1} = \mathbf{h}_l + f(\mathbf{h}_l), \quad (4)$$

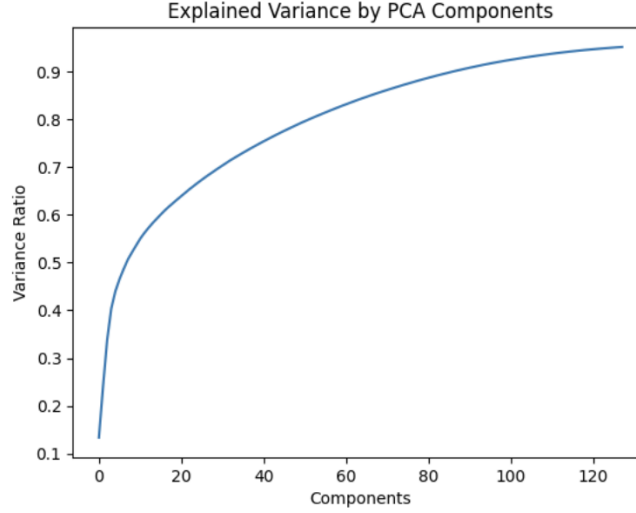


Figure 2: Cumulative explained variance for Task 1 embeddings. 128 components capture approximately 95% of total variance.

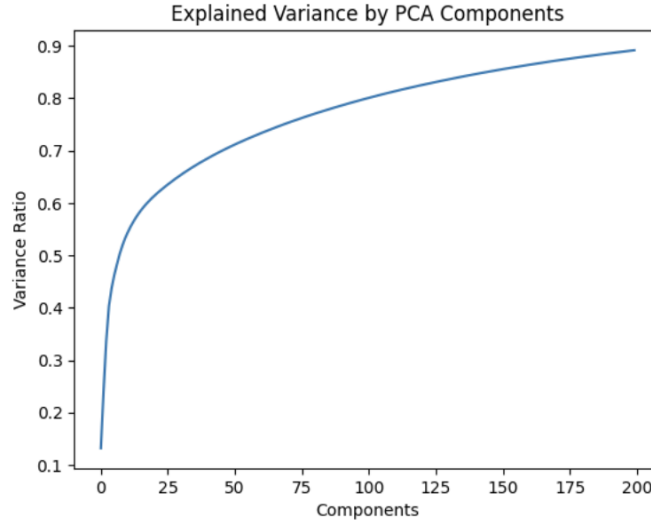


Figure 3: Cumulative explained variance for Task 2 embeddings. Variance saturates more gradually, motivating retention of 200 components.

where $f(\cdot)$ denotes a two-layer feedforward transformation with GELU activation and dropout (rate 0.1). GELU is selected instead of ReLU to provide smoother non-linearities and reduce the risk of dead neurons, that is, the “Dying ReLU” problem.

5. Implementation Details

Experiments were conducted using PyTorch and HuggingFace Transformers. All feature extraction is performed on frozen SVELA language models, ensuring that verification remains strictly model-agnostic and does not modify parameters established during unlearning (Figure 1).

The official SVELA training split is internally divided using stratified sampling with a validation fraction of 20%, preserving class distribution across *retain*, *forget*, and *test*. Random seeds are fixed for reproducibility. Dimensionality reduction (IPCA) and feature standardization are fitted exclusively on the training subset and reused for validation and test splits to prevent information leakage.

The residual classifier is trained for 40 epochs using the AdamW optimizer with learning rate 3×10^{-4}

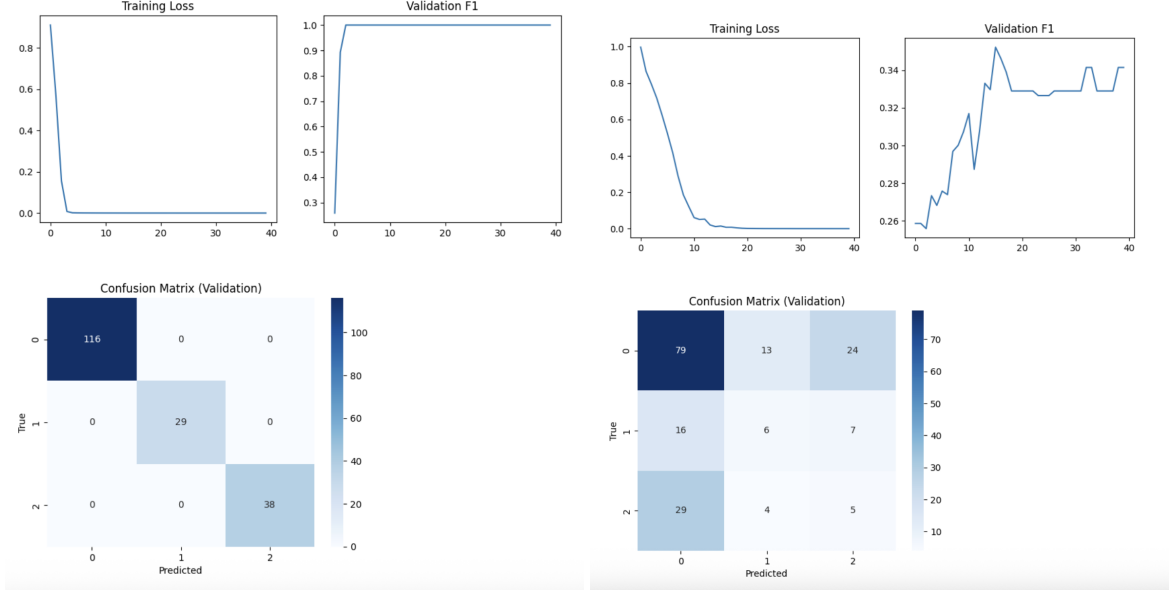


Figure 4: Training and validation performance. **Left:** Task 1 achieves nearly perfect separation within few epochs, reflecting its more distinct topic-based structure. **Right:** Task 2 shows slower convergence with class confusions, particularly between *retain* and *test*.

Table 4

Global hyperparameters shared across Task 1 and Task 2.

Parameter	Value
Optimizer	AdamW
Learning rate	3×10^{-4}
Loss function	Cross-Entropy
Classifier batch size	32
Hidden dimension	256
Number of residual blocks	2
Activation function	GELU
Dropout rate	0.1
Validation split	20% (stratified)
Random seed	42
Feature scaling	StandardScaler (train-only fit)

and batch size 32. The objective function is the multi-class cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^3 \mathbb{1}[y_i = c] \log \hat{p}_{i,c}, \quad (5)$$

where y_i denotes the ground-truth label and $\hat{p}_{i,c}$ represents the predicted probability for class c . Model selection is performed using macro-averaged F1 score on the validation set. No explicit class weighting is applied, although the dataset is imbalanced in favour of the majority “retain” class in order to avoid information loss.

Global optimization and architectural hyperparameters shared across both tasks are summarized in Table 4, while task-specific settings are reported in Table 5.

Figure 4 provides insights into the training behavior and classification performance of the residual network for Task 1 and Task 2. Task 1 exhibits near-perfect validation F1 after just a few epochs, and the confusion matrix confirms excellent class separation. In contrast, in Task 2, the validation F1 improves gradually before plateauing, and the confusion matrix indicates frequent misclassification between the *retain* and *test* classes, reflecting the subtle semantic differences in that setup. This suggests that Task 1,

Table 5

Task-specific hyperparameters for Task 1 and Task 2.

Parameter	Task 1	Task 2
Max sequence length	384	512
IPCA components	128	200
Token batch size (feature extraction)	8	8
Training epochs	40	40

Table 6

Task 1 leaderboard results (mean and standard deviation of macro-averaged F1 score).

Team	1B Models		3B Models	
	Mean	Std	Mean	Std
priyam_saha17	0.3428	0.0100	0.3565	0.0068
MALTO	0.3363	0.0091	0.3526	0.0083
ItaLib	0.3184	0.0054	0.3109	0.0037
matteoberta	0.3183	0.0365	0.3284	0.0437

Table 7

Task 2 leaderboard results (mean and standard deviation of macro-averaged F1 score).

Team	1B Models		3B Models	
	Mean	Std	Mean	Std
priyam_saha17	0.3345	0.0045	0.3335	0.0049
MALTO	0.3234	0.0076	0.3321	0.0049
matteoberta	0.2882	0.0131	0.2838	0.0047

which includes entity-level forgetting, provides clearer discriminative signals for the classifier, likely due to more distinct shifts in representation between retained and forgotten instances.

6. Results

The system ranked first on all official leaderboards across both Task 1 and Task 2, for both 1B and 3B model settings. Tables 6 and 7 report the aggregate mean macro-averaged F1 scores (with standard deviation) across all evaluated unlearning algorithms. Across both tasks, the approach achieves the highest average macro-F1 compared to other participating teams. Performance remains competitive across model sizes, with slight variations between the 1B and 3B settings depending on the task and unlearning strategy.

Although submissions were made on four selected base models (Section 4.1), the evaluation scripts were executed on both 1B and 3B parameter variants under all considered unlearning strategies: adversarial negative training (advneg), standard fine-tuning on the retain set (finetune), KL-divergence minimization (kl_min), negative gradient unlearning (neggrad), and preference optimization (pref_opt).

The detailed class-wise analysis (Tables 8 and 9) reveals consistent disparities across classes. Class 0 (“Retain”) generally achieves substantially higher precision, recall, and F1 scores compared to Classes 1 (“Forget”) and 2 (“Test”) across tasks, model sizes, and unlearning methods. In contrast, Classes 1 and 2 exhibit lower recall and F1 values, with performance varying more noticeably across unlearning strategies. This pattern suggests that the models retain stronger predictive performance for one class while exhibiting reduced separability or higher confusion for the remaining classes.

Table 8

Detailed results (Part 1): Overall score and class-wise accuracy for Task 1 and Task 2 (1B and 3B models).

Team	Task	Dim	Unlearner	Score	OverallAcc	C0Acc	C1Acc	C2Acc
priyam_saha17	task1	1b	advneg	0.3517	0.5100	0.6932	0.1528	0.2072
priyam_saha17	task1	1b	finetune	0.3540	0.5206	0.7143	0.1542	0.1910
priyam_saha17	task1	1b	kl_min	0.3379	0.4957	0.6780	0.1472	0.1887
priyam_saha17	task1	1b	neggrad	0.3300	0.4993	0.6932	0.1236	0.1771
priyam_saha17	task1	1b	pref_opt	0.3405	0.5084	0.7030	0.1458	0.1725
priyam_saha17	task1	3b	advneg	0.3508	0.5170	0.7073	0.1319	0.2141
priyam_saha17	task1	3b	finetune	0.3552	0.5138	0.6914	0.1208	0.2593
priyam_saha17	task1	3b	kl_min	0.3507	0.4941	0.6575	0.1514	0.2442
priyam_saha17	task1	3b	neggrad	0.3583	0.5138	0.6953	0.1750	0.2014
priyam_saha17	task1	3b	pref_opt	0.3673	0.5356	0.7295	0.1458	0.2245
priyam_saha17	task2	1b	advneg	0.3318	0.5035	0.6904	0.1264	0.1934
priyam_saha17	task2	1b	finetune	0.3345	0.5094	0.7018	0.1323	0.1934
priyam_saha17	task2	1b	kl_min	0.3253	0.4894	0.6680	0.1264	0.1812
priyam_saha17	task2	1b	neggrad	0.3234	0.4888	0.6737	0.1205	0.1763
priyam_saha17	task2	1b	pref_opt	0.3276	0.4941	0.6823	0.1264	0.1837
priyam_saha17	task2	3b	advneg	0.3299	0.5059	0.6932	0.1215	0.1837
priyam_saha17	task2	3b	finetune	0.3335	0.5100	0.7018	0.1323	0.1887
priyam_saha17	task2	3b	kl_min	0.3268	0.4894	0.6680	0.1323	0.1837
priyam_saha17	task2	3b	neggrad	0.3302	0.4976	0.6842	0.1264	0.1887
priyam_saha17	task2	3b	pref_opt	0.3321	0.5012	0.6904	0.1323	0.1887

Table 9

Detailed results (Part 2): Class-wise precision, recall and F1 scores for Task 1 and Task 2 (1B and 3B models).

Team	Task	Dim	Unlearner	Class 0			Class 1			Class 2		
				Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
priyam_saha17	task1	1b	advneg	0.6417	0.6932	0.6664	0.2249	0.1528	0.1820	0.2062	0.2072	0.2067
priyam_saha17	task1	1b	finetune	0.6455	0.7143	0.6782	0.2418	0.1542	0.1883	0.2005	0.1910	0.1956
priyam_saha17	task1	1b	kl_min	0.6368	0.6780	0.6567	0.1924	0.1472	0.1668	0.1918	0.1887	0.1902
priyam_saha17	task1	1b	neggrad	0.6388	0.6932	0.6649	0.1784	0.1236	0.1460	0.1813	0.1771	0.1792
priyam_saha17	task1	1b	pref_opt	0.6396	0.7030	0.6698	0.2201	0.1458	0.1754	0.1804	0.1725	0.1763
priyam_saha17	task1	3b	advneg	0.6493	0.7073	0.6770	0.2317	0.1319	0.1681	0.2009	0.2141	0.2073
priyam_saha17	task1	3b	finetune	0.6509	0.6914	0.6705	0.2308	0.1208	0.1586	0.2173	0.2593	0.2364
priyam_saha17	task1	3b	kl_min	0.6501	0.6575	0.6538	0.2084	0.1514	0.1754	0.2051	0.2442	0.2229
priyam_saha17	task1	3b	neggrad	0.6513	0.6953	0.6726	0.2291	0.1750	0.1984	0.2064	0.2014	0.2039
priyam_saha17	task1	3b	pref_opt	0.6641	0.7295	0.6953	0.2574	0.1458	0.1862	0.2163	0.2245	0.2203
priyam_saha17	task2	1b	advneg	0.6378	0.6904	0.6630	0.2075	0.1264	0.1574	0.1872	0.1934	0.1902
priyam_saha17	task2	1b	finetune	0.6443	0.7018	0.6718	0.2137	0.1323	0.1635	0.1887	0.1934	0.1910
priyam_saha17	task2	1b	kl_min	0.6322	0.6680	0.6496	0.2050	0.1264	0.1563	0.1784	0.1812	0.1798
priyam_saha17	task2	1b	neggrad	0.6357	0.6737	0.6542	0.2018	0.1205	0.1508	0.1756	0.1763	0.1760
priyam_saha17	task2	1b	pref_opt	0.6395	0.6823	0.6602	0.2094	0.1264	0.1579	0.1816	0.1837	0.1826
priyam_saha17	task2	3b	advneg	0.6412	0.6932	0.6661	0.2049	0.1215	0.1527	0.1835	0.1837	0.1836
priyam_saha17	task2	3b	finetune	0.6468	0.7018	0.6731	0.2145	0.1323	0.1637	0.1879	0.1887	0.1883
priyam_saha17	task2	3b	kl_min	0.6339	0.6680	0.6505	0.2086	0.1323	0.1617	0.1795	0.1837	0.1816
priyam_saha17	task2	3b	neggrad	0.6381	0.6842	0.6603	0.2107	0.1264	0.1580	0.1834	0.1887	0.1860
priyam_saha17	task2	3b	pref_opt	0.6426	0.6904	0.6656	0.2139	0.1323	0.1637	0.1868	0.1887	0.1877

7. Discussion

The strong results across both SVELA tasks show that combining embeddings with uncertainty-based features (like entropy and logit margin) works well for verifying forgetting. These different signals seem to capture complementary aspects of the model’s behavior, and using them together gives better performance than using any one signal alone.

Looking at the confusion matrices (Figure 4), most errors happen between the *retain* and *test* classes. This suggests that when forgetting is imperfect, the retained examples may still share some patterns with forgotten ones. Task 2 showed more of this confusion, likely because it involved removing individual facts (topics), which creates more overlap in the remaining data. In contrast, Task 1 dealt with entire identities, which gave the model cleaner separation between classes.

Further, using a fixed English prompt across all languages helped reduce variation in model output, which was useful for this multilingual setting. This strategy might help other multilingual evaluation tasks where consistent behavior is important.

Finally, Table 9 clearly shows that F1 scores for the “forget” and “test” classes are significantly lower than for the “retain” class across all unlearning strategies in both Task 1 and Task 2 (both 1B and 3B models). This suggests the classifier tends to overpredict the retain class which is likely due to its higher frequency and more consistent representations post-unlearning.

8. Limitations

While the proposed system attained top-ranked performance across all SVELA tracks, the absolute scores remain modest, with macro F1 hovering around 33–35%. This outcome highlights the inherent difficulty of selective forgetting verification and suggests that substantial headroom remains for future improvements.

The approach depends on access to labeled verification data, which may not be available in real-world unlearning deployments. Although the classifier is trained independently of the LLM parameters, the overall method still assumes the availability of structured prompt–response pairs and class annotations.

The system adopts several architectural choices that present opportunities for deeper exploration. For example, a shallow residual classifier is used with the assumption that residual connections support better gradient flow and feature fusion. While this design choice proved effective, comparative ablations with simpler MLP architectures were not conducted and could provide further clarity. Similarly, the dimensionality reduction stage relies on IPCA with empirically chosen component sizes based on validation performance; exploring alternative advanced methods such as autoencoders or manifold learning could offer additional benefits. As future work, advanced classifier architectures and richer feature extraction strategies may be incorporated to enhance generalization and interpretability across tasks.

Lastly, while class imbalance is handled by avoiding explicit downsampling of the “retain” class, the classifier displays a notable prediction bias towards this dominant class. The per-class F1 breakdown reveals persistent underperformance on the “forget” and “test” categories, which suggests that detecting forgotten content remains a challenging open problem.

9. Conclusion

This work presented a feature-based pipeline for verifying whether large language models have effectively forgotten targeted information following unlearning. The system relies on black-box access to LLMs and combines semantic embeddings with uncertainty and behavioral signals to train a lightweight classifier. All components are modular and task-agnostic, allowing the approach to generalize well across both entity-level and fact-level forgetting.

The method demonstrated consistent first-place performance across all four SVELA tracks, indicating the effectiveness of using fused representational and uncertainty features for this verification task. Despite its simplicity, the approach proved competitive and scalable across model sizes.

Future work could investigate unsupervised or semi-supervised strategies for verification, especially in settings where labeled forgotten examples are unavailable. Further ablation studies are also necessary to better understand the contribution of each component, including the effect of residual architecture and the role of individual uncertainty features. Broader evaluation across more diverse languages and unlearning algorithms could help stress-test the pipeline’s robustness and inform next-generation verification protocols.

Acknowledgments

The author thanks the SVELA organizers for providing the datasets, pretrained models, and evaluation

infrastructure, and the EVALITA 2026 committee for organizing the shared task.

Declaration on Generative AI

During the preparation of this work, the author used GPT-5.2 and Grammarly to assist with sentence-level grammar correction and stylistic polishing of the manuscript, and figma for building the flowchart representation of the pipeline. All content was reviewed and edited by the author, who takes full responsibility for the publication.

References

- [1] A. Ginart, M. Guan, G. Valiant, J. Zou, Making ai forget you: Data deletion in machine learning, NeurIPS (2019).
- [2] L. Bourtoutle, V. Chandrasekaran, C. A. Choquette-Choo, R. Jia, A. Travers, B. Zhang, D. Lie, N. Papernot, Machine unlearning, in: IEEE Symposium on Security and Privacy, 2021.
- [3] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, Ú. Erlingsson, et al., Extracting training data from large language models, in: USENIX Security Symposium, 2021.
- [4] D. Choi, D. Na, Towards machine unlearning benchmarks: Forgetting the personal identities in facial recognition systems, 2023. URL: <https://arxiv.org/abs/2311.02240>. arXiv:2311.02240.
- [5] P. Maini, Z. Feng, A. Schwarzschild, Z. C. Lipton, J. Z. Kolter, Tofu: A task of fictitious unlearning for llms, 2024. URL: <https://arxiv.org/abs/2401.06121>. arXiv:2401.06121.
- [6] R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, C. Finn, Direct preference optimization: Your language model is secretly a reward model, 2024. URL: <https://arxiv.org/abs/2305.18290>. arXiv:2305.18290.
- [7] J. Hayes, I. Shumailov, E. Triantafillou, A. Khalifa, N. Papernot, Inexact unlearning needs more careful evaluations to avoid a false sense of privacy, 2024. URL: <https://arxiv.org/abs/2403.01218>. arXiv:2403.01218.
- [8] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, Svela at evalita 2026: Overview of the selective verification of erasure from llm answers task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [9] C. Savelli, M. La Quatra, A. Koudounas, F. Giobergia, FAME: Fictional actors for multilingual erasure, in: Proceedings of the Fifteenth Language Resources and Evaluation Conference, European Language Resources Association, 2026.
- [10] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [11] D. Hendrycks, K. Gimpel, A baseline for detecting misclassified and out-of-distribution examples in neural networks, in: International Conference on Learning Representations, 2017.
- [12] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015. URL: <https://arxiv.org/abs/1512.03385>. arXiv:1512.03385.

Code and Reproducibility

To support transparency and reproducibility, the full implementation, training pipeline, and submission files used for both SVELA tasks are publicly available as Kaggle Notebooks:

- Task 1 notebook: <https://www.kaggle.com/code/priyamsaha17/svela-task-1>

- Task 2 notebook: <https://www.kaggle.com/code/priyamsaha17/svela-task-2>

A comprehensive GitHub repository containing all notebooks, trained models, evaluation scores, and performance curves is publicly available.¹

¹<https://github.com/priyam-saha-17/A-Feature-Centric-Pipeline-for-Verifying-Selective-Forgetting-in-Large-Language-Models>

EVWSD-ITA at EVALITA 2026: Overview of the Enhanced Visual Word Sense Disambiguation for Italian Task

Elio Musacchio^{1,2,*}, Lucia Siciliani¹, Pierpaolo Basile¹ and Giovanni Semeraro¹

¹Dept. of Computer Science, University of Bari Aldo Moro, Via E. Orabona, 4 - 70125 Bari (ITALY)

²Dept. of Computer Science, University of Pisa, Largo Bruno Pontecorvo, 3, Pisa (ITALY)

Abstract

EVWSD-ITA (Enhanced Visual Word Sense Disambiguation for Italian) is a shared task proposed at the EVALITA 2026 campaign. While traditional Visual Word Sense Disambiguation (VWSD) focuses on broad semantic distinctions, EVWSD-ITA introduces a more rigorous challenge by requiring systems to perform fine-grained disambiguation. The task involves selecting the most appropriate image for a target word within a given context, specifically designed to include "hard negatives"—co-hyponyms that share a common hypernym but represent distinct concepts. The dataset was meticulously constructed, comprising a manually validated test set. In this report, we showcase the dataset construction procedure and the results of the task.

Keywords

Word Sense Disambiguation, Visual Word Sense Disambiguation, Dataset, Evaluation

1. Introduction

Word Sense Disambiguation (WSD) is a historical task in the Natural Language Processing field of research [1]. In this task, the objective is to select the correct sense of a target word in a sentence, out of all its possible meanings. Given its longevity, several works have explored algorithms and neural networks capable of solving the task [2]. Nevertheless, this task still cannot be solved entirely, thus capturing all the nuances of natural language remains an intriguing challenge. The introduction of Large Language Models (LLMs) has further ignited interest in resolution of this task, with several works proposing usage of LLMs to perform WSD [3]. In light of the success and interest of WSD, Visual Word Sense Disambiguation (VWSD) [4] has been proposed. In this task, the objective is to select an image, out of ten possible candidates, which correctly represents the sense of a target word in a given sentence. The sentence consists of a word of interest (that is, the target to disambiguate) and additional context words to support disambiguation (e.g., "music keyboard", where "keyboard" is the target to disambiguate and "music" is the context word). The gold standard is represented by examples annotated manually, where human experts have selected images that are meaningful with respect to the text, exploiting BabelDomains from BabelNet. Furthermore, the original VWSD dataset is split into three languages: English, Italian and Farsi. However, the task has a very clear limitation: it does not consider fine-grained semantics. Considering the previous example, for the target word "music keyboard", any image not related to music can be discarded, which favors systems with a good understanding of high-level semantic relationships across language and vision. The task does not include recognition of specific senses (e.g., disambiguating the image of a piano from the image of a harpsichord). In light of this, we propose a new task that combines both high-level and fine-grained semantics. The goal is not only to identify the broad sense of the target word, but also to accurately recognise its specific sense. In light of this, we propose the Enhanced Visual Word Sense Disambiguation for Italian task (EVWSD-ITA) at EVALITA 2026 [5].

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ elio.musacchio@uniba.it (E. Musacchio); lucia.siciliani@uniba.it (L. Siciliani); pierpaolo.basile@uniba.it (P. Basile); giovanni.semeraro@uniba.it (G. Semeraro)

🆔 0009-0006-9670-9998 (E. Musacchio); 0000-0002-1438-280X (L. Siciliani); 0000-0002-0545-1105 (P. Basile); 0000-0001-6883-1853 (G. Semeraro)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The proposed task is significantly different from others in the state-of-the-art for the following reasons:

- We incorporate both high-level and fine-grained semantics by combining images representing different senses of the target word and co-Hyponyms (synsets that share the same Hypernym)
- We focus on the Italian language and manually annotate the test set to guarantee the quality and rigorousness of the benchmark

2. Related Works

The original WSD task has been extensively studied in previous research. In this context, one of the most used datasets is XL-WSD [6], a cross-lingual evaluation benchmark for the WSD task featuring sense-annotated development and test sets in 18 languages from six different linguistic families. Originally, WSD disambiguation models leveraged the BERT model. For example, Huang et al. [7] proposed GlossBERT, a family of models fine-tuned on context-gloss pairs and then converting WSD to a sentence-pair classification task. More recent approaches started to focus on leveraging LLMs for WSD. For example, Yae et al. [8] proposed three techniques to leverage LLMs in WSD. They format the task in the following formats: 1) multiple-choice questions; 2) binary questions; 3) multiple-choice questions for unseen words. Furthermore, Meconi et al. [9] propose an extensive study on the capabilities of LLMs in the WSD task, with a focus on the lexical understanding capabilities of the best performing models.

Following the success of WSD, the VWSD task was proposed for SemEval 2023 and, even after the completion of the challenge, several solutions and extensions of the dataset were proposed. Kritharoula et al. [10] tested several approaches using both generative LLMs for phrase enhancement and encoder-based vision-language models for retrieval. Specifically, the authors explore several possible solutions to address the VWSD task, they also consider image captioning and image retrieval, in order to cast the task as a unimodal retrieval task, as well as a learning to rank system and question-answering with chain-of-thought prompting. Kwon et al. [11] proposed an unsupervised VWSD approach exploiting definition generation with GPT-3 and Bayesian inference. Specifically, the authors perform context-aware definition generation with GPT-3 to overcome the out of vocabulary problem (that is, not all words having their definitions available in a lexical knowledge-base). Then they perform Bayesian style inference for Image-Text Matching. Zhang et al. [12] proposed SRCB, which employs a three stage pipeline: a context retrieval module, which predicts the correct definition using a bi-encoder architecture, a image retrieval module, which retrieves the relevant images from an image dataset, and a matching module, which decides to use either text or images and ranks them. Yang et al. [13] proposed MTA, which employs self-distillation to align fine-grained textual features to fixed vision features and align non-English textual features to English textual momentum features. Furthermore, they introduced a trilingual image-text dataset for VWSD, encompassing a fine-grained network of 85,754 word-sense associations and 120,131 images. Laba et al. [14] extended the original dataset to include the Ukrainian language. The authors extended the dataset through a semi-supervised approach leveraging Wikipedia articles and expert annotators. Furthermore, they test the performance of eight multilingual and multimodal LLMs using this dataset. Setitra et al. [15] generated visually representative images from textual descriptions, as well as rich textual descriptions from images. Then, an ensemble of deep models is used to perform classification. Musacchio et al. [16] proposed to use strictly hard negative samples exploiting the co-Hyponym relation from semantic networks. The authors then propose a new benchmark on a dataset created using this strategy and extensively evaluate both Vision-Language Encoders and Large Language Models supporting multimodal inputs.

Despite the research interest and popularity of the task, no work considers the proposal of a benchmark that combines high-level and fine-grained semantics on this task.

3. Task Description

We propose a single task, that is Visual Word Sense Disambiguation. Given a query and ten possible candidate images for it, we want to learn a system that is capable of selecting the relevant images for the query. In the task, only one candidate image is relevant for the query. Hence, the system must be able to either: 1) predict the one relevant image; 2) provide a score for each image based on its relevance to the query.

In EVWSD-ITA, we combine two different types of images: 1) images representing a different sense of the target word, for example the word "keyboard" may be paired to both the image of a computer keyboard and the image of a piano; 2) images representing a different specific sense for the target word while sharing the same broad sense, for example the word "keyboard" may be paired to both the image of a piano and the image of a harpsichord. We illustrate this in Figure 1. We use co-Hyponyms extracted from a semantic network to find *hard negatives* [17, 16]. That is, the images are related to the general sense of the target word, but that represent a different specific sense. We propose to improve the task by mixing the two types of images: 1) images for other senses of the target word; 2) images that share the same broad sense as the target word. This will test the ability of the proposed solutions to understand both fine-grained and high-level semantics.

In the original VWSD challenge, two metrics were used to assess the ability of the proposed solutions: HIT@1 and MRR (Mean Reciprocal Rank). It may be impossible to compute MRR in some cases, since it expects a ranking (e.g., systems trained for prediction only rather than ranking cannot be benchmarked using MRR). Hence, we will consider two different leaderboards: one focusing on MRR and the other focusing on HIT@1. Thanks to this, we can distinguish the goodness of the proposed solution on two different aspects (precise classification and overall ranking quality). We will describe the metrics in detail in Section 5.1.

4. Dataset

In order to propose the EVWSD-ITA task, we need a dataset where images satisfy the properties we are interested in (images representing synsets from that co-Hyponym of the target and synsets that share the same lemma as the target).

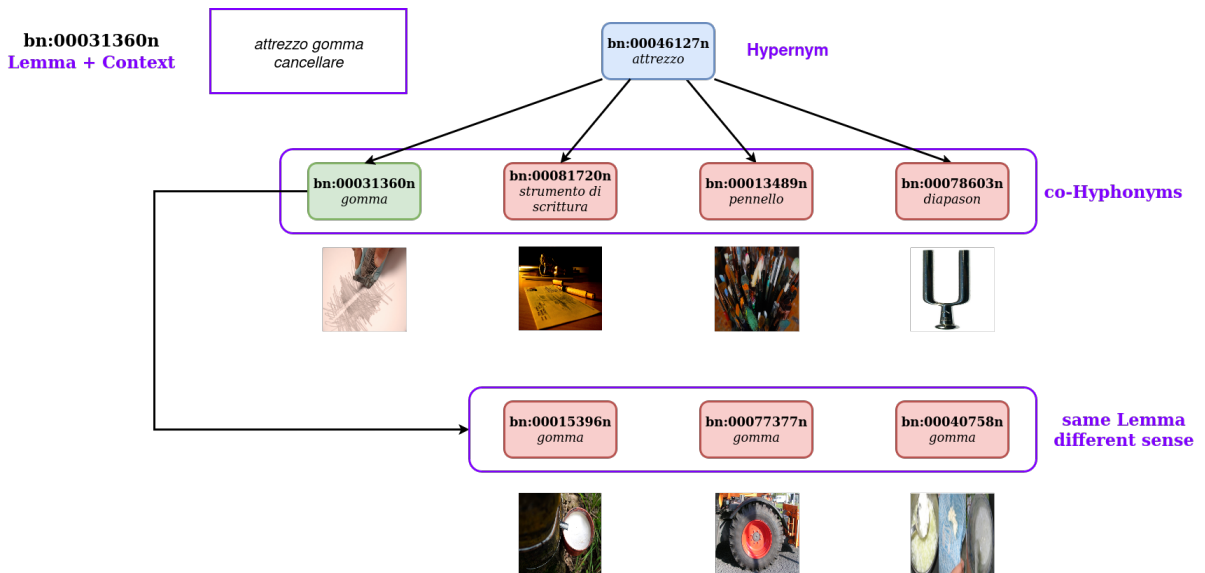


Figure 1: Example of the proposed approach. The synset in green is the correct one, while the ones in red are negatives.

This section describes the creation of the dataset and provides quantitative statistics.

4.1. Dataset Construction

Our data collection relies on BabelNet [18]. Since it provides images for each synset, it is a valuable resource to obtain data for our task. For each synset in BabelNet with an English gloss, we extracted the synset identifier and the corresponding Italian gloss. Since a synset can have multiple glosses, we select only one gloss according to its source. In particular, for selecting only one gloss, we prioritised each source according to this rank: WordNet, Wikitionary, Wikipedia, Wikidata, and other sources. Then, the URL's image from BabelNet is extracted for each synset. Since more images can be assigned to each synset, we used the image tagged with the attribute "best image". Each synset $s_i \in S$, where S is the set containing all synsets in BabelNet, has multiple lemmas $L(s_i) = \{l_1, \dots, l_{m_i}\}$, where m_i is the number of lemmas for synset s_i , and a gloss g_i associated to it. We define a relationship $Hyp : S \rightarrow S$ which connects each synset to its Hyponym. Furthermore, we define two sets associated to s_i , the co-Hyponyms set $CH(s_i) = \{s_j \mid Hyp(s_j) = Hyp(s_i) \wedge s_j \neq s_i \wedge s_j \in S\}$ and the set of synsets with the same lemma $SL(s_i, l_i) = \{s_j \mid \exists l_j \in L(s_j) : l_j = l_i \wedge s_j \neq s_i \wedge s_j \in S\}$.

For the test set, we created the data using the following methodology: we iterated over all lemmas in the collection and iterated over all possible synsets for each lemma. We discarded instances where there were no synsets in the co-Hyponyms set or in the synsets with the same lemma. Then, we manually checked each instance and the candidate images of each synset. We removed synsets from the candidates when: 1) the distinct synset specific sense was too similar to that of the target sense; 2) visual elements of the image associated to the synset were not enough to distinguish it with respect to the target synset. Specifically, for the first case, we found some synsets in BabelNet that shared the same gloss or represented the same sense as the target synset but with a different Hyponym. For the second case, we found cases where distinct synsets were represented using the same image or there were not enough visual elements to distinguish it with respect to the image of the target synset (e.g., two nearly identical electrical appliances without any distinguishing visual characteristic). We also removed instances where the semantic category of the target synset represents an abstract concept (e.g., "year" may be ambiguous to disambiguate through visual features). If after removing the synsets there were less than ten candidate synsets, we marked the instance to be augmented. After completing the test set manual validation, we augmented all instances with less than ten candidate synsets with random synsets selected from the other instances. Furthermore, we also manually created the query. To create the query, we combined: the lemma of the correct synset (the one we iterated over), one of the lemmas of the hypernym of the correct synset, and a word from the gloss of the correct synset. Using this methodology, we are sure that there is enough information to properly disambiguate the correct image. Furthermore, we randomize the order of each part of the query, in order to avoid participants being able to reverse engineer the construction of the query and obtaining information they should not have access to (without randomizing the order the first word of the query would always be the lemma of the correct synset). The word from the gloss should help disambiguate fine-grained semantics, while the lemma from the hypernym the high-level sense of the input. To increase the degree of diversity of the test split with respect to the train split, we also performed additional steps. We included synsets where glosses in Italian are not present. When using that synset as target, we manually translated the gloss from the original English one to Italian and selected the word to be used in the query. Furthermore, for all target synsets, we manually replaced the image from BabelNet with a different image sourced from sites with a permissive license.

For the train set, we do not provide a query directly. While creating the query using the lemmas of the target synset and its hypernym is easy, selecting a meaningful word from the gloss associated to the synset is not trivial. Participants are asked to develop a strategy to extract the word from the gloss to help with disambiguation. This also allows participants to have more freedom to use the dataset as they see fit, as long as the final model is capable of performing VWSD given only a query and a candidate pool of images.

4.2. Dataset Statistics

Listing 1: Example of Train Instance

```
1 {
2   "id": "bn:00022412n",
3   "hyp_id": "bn:00017670n",
4   "gloss": "Atto del cuocere",
5   "lemma": "cucina",
6   "hyp_lemma": [
7     "cambiamento di stato"
8   ],
9   "bns": [
10    "bn:00018237n", ..., "bn:00049248n"
11  ],
12  "is_co_hyp": [
13    true, ..., false
14  ],
15  "images": [
16    "F14/bn:00018237n", ..., "F0/bn:00049248n"
17  ],
18  "all_lemmas": [
19    ["masticazione", "masticare"],
20    ...,
21    ["cucine", "cucina", "cucina attrezzata", "cucina aperta", "cucinotto"]
22  ],
23  "all_glosses": [
24    "La masticazione è il processo mediante il quale il cibo è frantumato e preparato dai denti.",
25    ...,
26    "Una stanza attrezzata per la preparazione dei cibi."
27  ],
28  "img": "F22/bn:00022412n"
29 }
```

For the train split, we provide a total of 10,000 instances. These instances are not formatted for the task, we provide the data of interest from BabelNet directly. Specifically, we provide for each instance the following fields:

- **‘id’**: Synset id
- **‘hyp_id’**: Synset id for the hypernym
- **‘gloss’**: Gloss in Italian
- **‘lemma’**: Lemma in Italian
- **‘hyp_lemma’**: All possible lemmas for the hypernym
- **‘bns’**: The list will contain both: co-hyponyms, synsets that have the "lemma" as a possible lemma
- **‘is_co_hyp’**: True if the corresponding synset in "bns" is a co-hyponym, False otherwise
- **‘images’**: Path to the image for each synset in "bns"
- **‘all_lemmas’**: Each list contains all the Italian lemmas associated to each synset in "bns"
- **‘all_glosses’**: All Italian glosses associated to each synset in "bns"
- **‘img’**: Path to the image

Each instance is a unique synset in BabelNet. Participants are asked to format the dataset leveraging the provided data. We decided to implement this approach to let participants have complete freedom over the data and how they wanted to format the train set. For example, using this dataset to train

Listing 2: Example of Test Instance

```

1 {
2   "query": "patologia calcolo organo",
3   "candidates": [
4     "1133.jpg",
5     "850.jpg",
6     "743.jpg",
7     "1266.jpg",
8     "1367.jpg",
9     "948.jpg",
10    "549.jpg",
11    "695.jpg",
12    "1504.jpg",
13    "1148.jpg"
14  ],
15 }

```

a CLIP model (vision-language encoder) would require a different formatting with respect to a LLM supporting multimodal inputs. A complete example of train instance is reported in Listing 1.

For the test split, we provide a total of 222 instances, following the methodology previously described. Furthermore, image paths have been anonymized (since in the train split the image path was matched to the synset id of the instance) in order to avoid participants being able to extract additional information from Babel synsets at test time. A complete example of test instance is reported in Listing 2.

For both the train and test splits, all images are provided in a square size of 336. This is done in order to provide participants with the same experimental setting for visual inputs. Furthermore, this allows easier use of Large Language Models supporting multimodal inputs, since they treat visual inputs as sequences of tokens, smaller images are processed more efficiently with respect to higher resolutions.

5. Evaluation

5.1. Metrics

We will evaluate the models using the two metrics considered in the original challenge: HIT@1 and MRR. Given $r = [r_1, \dots, r_n]$, where n is the cardinality of the test set and r_i is the rank of the correct image given as output by the model, MRR is defined as:

$$MRR = \frac{1}{n} \sum_{i=1}^n \frac{1}{r_i} \quad (1)$$

This metric is used to evaluate the goodness of the ranking; the closer r_i is to 1 (i.e., the first position in the ranking), the better the result. HIT@1 is defined as:

$$HIT@1 = \frac{1}{n} \sum_{i=1}^n I(r_i) \quad (2)$$

where I is a function that returns 1 if $r_i == 1$ (i.e., the correct image is ranked first), and 0 otherwise. Therefore, this metric assesses the model's ability to select the correct image as the best possible candidate.

5.2. Baseline

We use the same baseline that was used in VWSD for SemEval 2023, that is a multilingual CLIP model¹ trained using the sentence-transformers library [19]. We do not perform additional fine-tuning of the model, we use the pre-trained model directly. We extract the embedding for the query text in the multimodal embedding space of the multilingual CLIP model, then we extract the embeddings for all candidate images for that query. We compute the cosine similarity for the query embedding with respect to all image embeddings and rank the candidate images in descending order with respect to cosine similarity (image with highest similarity at the top of the list). After obtaining the ranked list, we can compute HIT@1 and MRR. This method provides us with a strong baseline that doesn't include additional information from external sources or ensemble models.

5.3. Participants

One team - UniTor (University of Tor Vergata) - participated in the task and submitted three distinct runs. In this section, we briefly describe the system proposed by said participants.

5.3.1. UniTor

The team proposed a five-stage pipeline: semantic analysis and weighted binary question generation, visual assessment, visual question-answering confidence estimation, multimodal semantic matching, multi-channel fusion and ranking. The final scoring used in the ranking is computed using different scores obtained in the pipeline. Specifically, the scoring combines a visual-question answering score and a semantic similarity score obtained from a CLIP model. Semantic analysis and question generation is done by using a LLM (GPT-5.1). The visual-question answering score is obtained by leveraging the probability of generating a "Yes/No" answers of a smaller LLM (Qwen3-VL), while the similarity score is obtained by leveraging a textual description w.r.t. candidate image. We define the scoring methodology to report the results as follows:

- Score VQA: visual question-answering score obtained by leveraging the probability of generating a "Yes/No" answer;
- Score VQA (Two-Step): visual question-answering score obtained by leveraging the probability of generating a "Yes/No" answer. The questions for this score are generated using a more strict parsing approach;
- Score CLIP (Query): similarity score obtained by a CLIP model using the candidate image and the original query;
- Score CLIP (Description): similarity score obtained by a CLIP models using the candidate image and the description generated by a LLM.

Complete details regarding the participants solutions are detailed in their report [20].

6. Results

We report results for HIT@1 in Table 1 and for MRR in Table 2. Overall the second system proposed by UniTor performed better than all others in both the HIT@1 and MRR leaderboards. Still, the first proposed system performed remarkably well in terms of MRR (.8100 against the .8182 of the best system). This highlights that the first system is still capable of providing good rankings with respect to the best system. Furthermore, we highlight that all proposed systems by UniTor significantly outperformed the baseline. This highlights that their proposed solution was capable of overcoming a system based on pure semantic similarity scoring. Finally, the system proposed by UniTor leveraged Large Language Models supporting multimodal inputs. This highlights current interest in multimodal research, where researchers are not strictly interested in extraction of embeddings from vision-language

¹<https://huggingface.co/sentence-transformers/clip-ViT-B-32-multilingual-v1>

encoder models, but are interested in leveraging Large Language Models supporting multimodal inputs for task resolution.

Team	Model	Score
baseline	clip-ViT-B-32-multilingual-v1	.4505
UniTor	Score VQA + Score CLIP (Query)	.6937
UniTor	Score VQA + Score CLIP (Query) + Score CLIP (Description)	.7117
UniTor	Score VQA (Two-Step) + Score CLIP (Query) + Score CLIP (Description)	.6667

Table 1
HIT@1 Leaderboard

Team	Model	Score
baseline	clip-ViT-B-32-multilingual-v1	.6244
UniTor	Score VQA + CLIP (Query)	.8100
UniTor	Score VQA + CLIP (Query) + CLIP (Description)	.8182
UniTor	Score VQA (Two-Step) + CLIP (Query) + CLIP (Description)	.7770

Table 2
MRR Leaderboard

7. Conclusions

In EVWSD-ITA, we proposed the first dataset for VWSD for fine-grained and high-level semantics in Italian. Given a query containing a target word and additional context, we combine both images representing the same sense as the target word (high-level semantics) and hard negatives obtained by exploiting the co-Hyponyms relation (fine-grained semantics). We provide an extensive train set obtained from BabelNet and a manually annotated test set. The EVWSD-ITA task was approached by a single team that proposed a solution based on Visual Question-Answering and semantic similarity obtained from a CLIP model. All participants' solutions were able to surpass the proposed baseline, for both metrics considered in this task (HIT@1 and MRR), highlighting the impact of their scoring strategy in VWSD.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] R. Navigli, Word sense disambiguation: A survey, ACM computing surveys (CSUR) 41 (2009) 1–69.
- [2] M. Bevilacqua, T. Pasini, A. Raganato, R. Navigli, Recent trends in word sense disambiguation: A survey, in: International joint conference on artificial intelligence, International Joint Conference on Artificial Intelligence, Inc, 2021, pp. 4330–4338.

- [3] D. Sumanathilaka, N. Micallef, J. Hough, Assessing gpt’s potential for word sense disambiguation: A quantitative evaluation on prompt engineering techniques, in: 2024 IEEE 15th Control and System Graduate Research Colloquium (ICSGRC), IEEE, 2024, pp. 204–209.
- [4] A. Raganato, I. Calixto, A. Ushio, J. Camacho-Collados, M. T. Pilehvar, SemEval-2023 task 1: Visual word sense disambiguation, in: A. K. Ojha, A. S. Doğruöz, G. Da San Martino, H. Tayyar Madabushi, R. Kumar, E. Sartori (Eds.), Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 2227–2234. URL: <https://aclanthology.org/2023.semeval-1.308/>. doi:10.18653/v1/2023.semeval-1.308.
- [5] F. Cutugno, A. Miaschi, A. P. Apro시오, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [6] T. Pasini, A. Raganato, R. Navigli, Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 13648–13656.
- [7] L. Huang, C. Sun, X. Qiu, X.-J. Huang, Glossbert: Bert for word sense disambiguation with gloss knowledge, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3509–3514.
- [8] J. H. Yae, N. C. Skelly, N. C. Ranly, P. M. LaCasse, Leveraging large language models for word sense disambiguation, Neural Computing and Applications 37 (2025) 4093–4110.
- [9] D. Meconi, S. Stirpe, F. Martelli, L. Lavalle, R. Navigli, Do large language models understand word senses?, in: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, 2025, pp. 33885–33904.
- [10] A. Kritharoula, M. Lymperaious, G. Stamou, Large language models and multimodal retrieval for visual word sense disambiguation, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 13053–13077. URL: <https://aclanthology.org/2023.emnlp-main.807/>. doi:10.18653/v1/2023.emnlp-main.807.
- [11] S. Kwon, R. Garodia, M. Lee, Z. Yang, H. Yu, Vision meets definitions: Unsupervised visual word sense disambiguation incorporating gloss information, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 1583–1598. URL: <https://aclanthology.org/2023.acl-long.88/>. doi:10.18653/v1/2023.acl-long.88.
- [12] X. Zhang, T. Zhen, J. Zhang, Y. Wang, S. Liu, Srcb at semeval-2023 task 1: Prompt based and cross-modal retrieval enhanced visual word sense disambiguation, in: Proceedings of the 17th international workshop on semantic evaluation (SemEval-2023), 2023, pp. 439–446.
- [13] Q. Yang, X. Wang, Y. Li, L.-K. Lee, F. L. Wang, T. Hao, Mta: A lightweight multilingual text alignment model for cross-language visual word sense disambiguation, in: ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2024, pp. 12166–12170.
- [14] Y. Laba, Y. Mohytych, I. Rohulia, H. Kyrleyza, H. Dydyk-Meush, O. Dobosevych, R. Hryniv, Ukrainian visual word sense disambiguation benchmark, in: Proceedings of the Third Ukrainian Natural Language Processing Workshop (UNLP)@ LREC-COLING 2024, 2024, pp. 61–66.
- [15] I. Setitra, P. Rajapaksha, A. K. Myat, N. Crespi, Leveraging ensemble deep models and llm for visual polysemy and word sense disambiguation, Multimedia Tools and Applications (2025) 1–33.
- [16] E. Musacchio, L. Siciliani, P. Basile, G. Semeraro, Assessing and improving the multilingual visual word sense disambiguation ability of vision-language models, in: ECAI 2025, IOS Press, 2025, pp. 4145 – 4152.
- [17] J. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, arXiv preprint arXiv:2010.04592 (2020).

- [18] R. Navigli, S. P. Ponzetto, Babelnet: Building a very large multilingual semantic network, in: Proceedings of the 48th annual meeting of the association for computational linguistics, 2010, pp. 216–225.
- [19] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: <http://arxiv.org/abs/1908.10084>.
- [20] C. D. Hromei, A. Scaiella, D. Croce, R. Basili, Unitor at evwsd-ita: Zero-shot visual word sense disambiguation via visual question-answering, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

UniTor at EVWSD-ITA: Zero-Shot Visual Word Sense Disambiguation via Visual Question-Answering

Claudiu D. Hromei¹, Antonio Scaiella², Danilo Croce¹ and Roberto Basili¹

¹Department of Enterprise Engineering, University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy

²Reveal S.r.l. - Via Kenia 21, 00144, Rome, Italy

Abstract

Visual Word Sense Disambiguation (VWSD) requires selecting, among ten candidates, the image that matches the intended sense of an ambiguous word given minimal context. We present the UniTor system for EVWSD-ITA at EVALITA 2026, a fully zero-shot generate-and-verify pipeline that combines: (i) LLM-based generation of discriminative positive/negative visual questions and a holistic target description, (ii) VQA-based verification over each candidate image with token-level confidence aggregation to score local visual evidence, and (iii) CLIP-based global matching against both the original query and the synthetic description. The final ranking is obtained by fusing local verification and global similarity signals. Experiments on the official benchmark show that our approach substantially improves over a CLIP-only baseline and is particularly effective at filtering hard negatives such as co-hyponyms and alternative senses.

Keywords

Visual WSD, Multi Modality, LLMs, Zero-Shot VQA

1. Introduction

The interaction between Natural Language and Visual Perception represents one of the long-standing goals of Artificial Intelligence [1, 2]. While humans effortlessly map linguistic concepts to visual objects, a process known as *symbol grounding* [3], machines often struggle when language becomes ambiguous. A word like “bank” can refer to a financial institution or a riverside; similarly, in Italian, the word “gomma” can denote an eraser, a car tyre, or simply a material. Visual Word Sense Disambiguation (VWSD) addresses this challenge: given a target word and a limited linguistic context, the system must identify the correct visual depiction among a set of candidates [4].

This paper describes the participation of the **UniTor** team in the **EVWSD-ITA** task at EVALITA 2026 [5]. The task extends the VWSD formulation (as introduced at SemEval2023 [6]) by introducing a high degree of visual complexity. Systems are provided with a target word, a hypernym, and a gloss fragment derived from BabelNet [7], and must retrieve the correct image from a pool of 10 candidates. Crucially, the candidate set includes not only unrelated images but also *hard negatives*: images representing different senses of the same word (polysemy) or visually similar objects from the same category (co-hyponyms).

Consider the example in Figure 1. The input query is “attrezzo gomma cancelleria” (en: “*tool rubber stationery*”). A standard dual-encoder model like CLIP [8], which relies on global visual-semantic similarity, might correctly discard a “tyre” (polysemous distinction) but fail to distinguish the “eraser” from a “pencil” or “ruler” (co-hyponyms), as they all share the visual embedding space of “stationery objects”. Most state-of-the-art approaches tackle this by fine-tuning massive encoders on task-specific data [9, 4]. However, this requires significant computational resources and often lacks interpretability.

We propose a different direction: a fully **Zero-Shot**, training-free architecture that re-frames disambiguation as a *Visual Constraint Verification* process. Inspired by recent advancements in Multimodal

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 - 27, Bari, IT
✉ hromei@ing.uniroma2.it (C. D. Hromei); scaiella@revealsrl.it (A. Scaiella); croce@info.uniroma2.it (D. Croce); basili@info.uniroma2.it (R. Basili)

ORCID 0009-0000-8204-5023 (C. D. Hromei); 0000-0001-9111-1950 (D. Croce); 0000-0001-5140-0694 (R. Basili)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

gomma attrezzo cancelleria

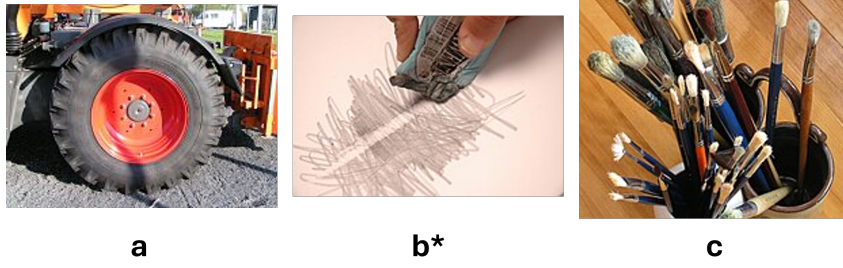


Figure 1: Running Example of the EVWSD-ITA Task. The input query is the scrambled triple “attrezzo gomma cancelleria” (en: “tool, rubber, stationery”). For simplicity, we report here only 3 images instead of 10 as in the task. The system must distinguish the *Target Image* (it:gomma, en:rubber, i.e. an eraser) from a *Polysemous Distractor* (a tyre) and a *Hard Negative Co-hyponym* (a pencil). Standard VLM embeddings often confuse the Target with the Hard Negative due to visual similarity. The Gold Standard image is noted with *.

Large Language Models (MLLMs) [10, 11, 12], our pipeline synergises the reasoning capabilities of LLMs with the discriminative power of Visual Question Answering (VQA). Instead of simply embedding the query, we use an LLM (GPT-5.1) to “explode” the ambiguous input into a set of discriminative requirements, formulated as Positive and Negative questions. A VQA model, specifically Qwen3-VL-8b [13], then acts as an auditor, verifying these constraints against each candidate image.

As an example, a simple way of distinguishing the *eraser* (image b from Figure 1) from the *tyre* (image a from Figure 1) would be asking a question that addresses its usage, such as “*L’oggetto è un grande anello nero con battistrada montato su una ruota?*” (en: “Is the object a large black ring with treads mounted on a wheel?”). On the other hand, in order to discriminate a hard negative, such as the *brushes* (image c from Figure 1), a very precise question, such as “*L’interazione primaria con questo oggetto prevede la rimozione di tratti grafici esistenti?*” (en: “Does the primary interaction with this object involve removing existing graphic features?”) will address the key difference between the two.

Our contribution is threefold: (1) We introduce a novel pipeline that disentangles linguistic reasoning from visual perception; (2) We demonstrate that generated “Negative Questions” are crucial for filtering hard negatives; (3) We show that this approach achieves competitive performance without any parameter updates. Specifically, we address the following Research Questions (RQs):

1. **RQ1:** Can the EVWSD problem be effectively modelled as a *Visual Constraint Verification* task, where the final ranking depends on the distinct satisfaction of linguistic requirements (“accounting”) rather than just embedding similarity?
2. **RQ2:** Can LLM-generated questions effectively filter hard negatives (e.g., co-hyponyms) in a zero-shot setting, surpassing standard global matching?
3. **RQ3:** Does the injection of a synthetic holistic description (generated by the LLM) provide a more robust context for global matching compared to the raw query?
4. **RQ4:** How does the ensemble of local feature verification (VQA) and global semantic matching (CLIP) impact the final ranking?

In the remainder of this paper, Section 2 reviews the State of the Art in Visual Word Sense Disambiguation, focusing on previous shared tasks and the limitations of current multimodal encoders. Section 3 details our proposed methodology, describing the five-stage pipeline from semantic analysis to multi-view fusion. Section 4 presents the experimental setup, reports the results achieved on the EVWSD-ITA dataset, and discusses the ablation study concerning the structured query parsing. Finally, Section 5 draws conclusions and outlines future research directions.

2. State of the Art

The task of Word Sense Disambiguation (WSD) has long been a cornerstone of Natural Language Processing, aiming to identify the correct meaning of a polysemous word within a given context. With the advent of large-scale multimodal knowledge bases such as **BabelNet** [7], this challenge has naturally extended to the visual domain. In BabelNet, concepts are organised into *synsets* (sets of synonyms sharing a common meaning), which are often associated with visual depictions. In this landscape, Visual Word Sense Disambiguation (VWSD) requires a system to map a linguistic query to the image that best represents the corresponding synset, distinguishing it not only from completely unrelated concepts but, more critically, from *co-hyponyms*, distinct concepts that share the same hypernym (ancestor) and possess similar visual features.

The complexity of this task was formally benchmarked in the **SemEval-2023 Task 1** on Visual Word Sense Disambiguation [4]. The shared task highlighted that the most successful approaches relied heavily on massive ensembles of Vision-Language Models (VLMs) or on fine-tuning state-of-the-art encoders like CLIP [8] on task-specific data. While effective, these strategies require significant computational resources and extensive training data, often overfitting to the specific visual domains of the training set (e.g., ImageNet-like pictures).

A significant limitation of standard dual-encoder models, such as CLIP, is their tendency to function as “visual bag-of-words” models. Research has shown that while these models excel at global semantic matching, they struggle with compositional understanding and syntax [14]. For instance, CLIP might align the query “gomma” (eraser) with an image of a “pencil” simply because both frequently appear in the context of “stationery”, failing to capture the fine-grained visual constraints that distinguish the two objects. This lack of granular reasoning makes them prone to errors when facing hard negatives, such as co-hyponyms.

To mitigate these limitations, recent works in Multimodal Learning have explored the use of Large Language Models (LLMs) to enhance visual reasoning. Techniques such as *Chain-of-Thought* (CoT) prompting [15] have been adapted to generate intermediate reasoning steps or to expand short queries into detailed visual descriptions [16, 17, 18, 19]. Similarly, Visual Question Answering (VQA) [20] has been proposed not just as a standalone task, but as a proxy to probe specific visual attributes in zero-shot classification scenarios.

Our approach distinguishes itself from the existing literature by integrating these components into a unified **Zero-Shot** pipeline that treats the input query not as a flat text string, but as a composite semantic object. Unlike previous methods that directly map the noisy query to the image space, we leverage the LLM to disentangle the semantic signals (lemma, hypernym, and gloss) and generate explicit verification constraints. By combining the generative power of LLMs with the discriminative precision of VQA, we address the lack of fine-grained reasoning in standard VLMs without the need for resource-intensive fine-tuning.

3. Visual Word Sense Disambiguation through Visual Question Answering

Our approach to the EVWSD-ITA task is grounded in the intuition that visual disambiguation requires a verify-then-match strategy. Rather than relying solely on global embedding similarities, we decompose the complex linguistic input into granular visual constraints. The proposed architecture is a five-stage pipeline that synergises the generative reasoning of Large Language Models (LLMs) with the discriminative verification of Vision-Language Models (VLMs). Figure 2 illustrates the five-stage pipeline adopted in our approach. The process begins with the **Input**, consisting of the scrambled linguistic triple and the set of candidate images. In **Phase 1**, the Large Language Model analyses the query to produce a structured JSON object containing a *Holistic Description* and a set of *Positive and Negative Questions*. The workflow then splits into two parallel branches:

- The **Local Verification Branch** (Phases 2 & 3), where the VQA model answers the generated

questions against the candidate images, producing the confidence score S_{VQA} via geometric mean aggregation.

- The **Global Matching Branch** (Phase 4), where the CLIP encoder computes the similarity of the image against both the original query ($S_{Original}$) and the generated holistic description ($S_{Synthetic}$).

Finally, in **Phase 5**, these three distinct signals are fused to compute the final score used to rank the images.

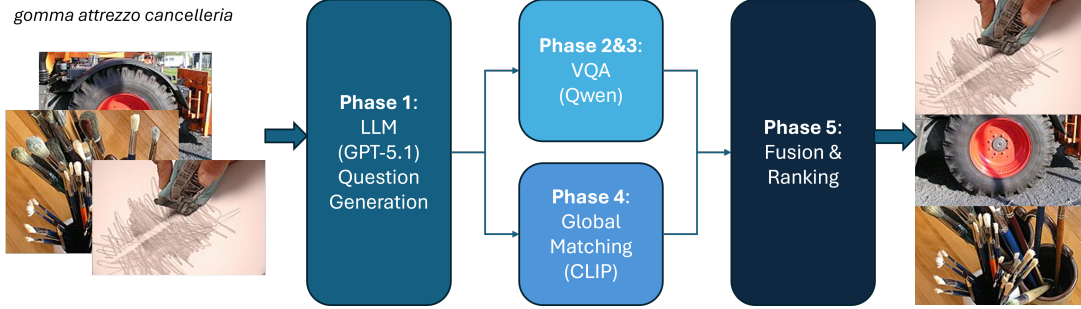


Figure 2: Overall workflow of the UniTor system. The pipeline processes the input query through an LLM (Phase 1) to generate discriminative questions and a holistic description. These outputs condition the visual verification performed by the VQA model (Phase 2 & 3) and the global matching performed by CLIP (Phase 4). Finally, the local and global scores are fused to produce the ranking (Phase 5).

3.1. Phase 1: Semantic Analysis and Weighted Question Generation

The first phase aims to disambiguate the input context via explicit semantic analysis to generate targeted visual checks. The raw input, a scrambled sequence of a lemma, a hypernym, and a gloss fragment (“gomma attrezzo cancelleria”), contains noisy signals that must be disentangled before visual matching can occur, i.e. we must first identify which is the main concept to focus on.

We employ **GPT-5.1** acting as a “*Computational Linguist*” to analyse this input fragment. Through a structured prompt, the model is instructed to perform a key reasoning step: identify the concrete *Visual Target* (the specific synset, i.e. “gomma”) implied by the gloss (i.e. “cancelleria”) and explicitly list potential *Semantic Distractions*, including specific co-hyponyms (e.g., distinguishing a “pencil” from an “eraser”) and alternative polysemous senses (e.g., distinguishing a “car tyre” from an “eraser”).

The output is a structured JSON object containing:

1. **Target Definition:** A synthetic, holistic description of the identified visual concept, which serves as a clean, expanded query for global matching. For the example in Figure 1: “*Piccolo oggetto solido in materiale gommoso o vinilico, impugnabile e sfregato sulla carta per rimuovere segni di matita o, in alcuni casi, di penna.*” (en: “A small, solid object made of rubber or vinyl that can be held in the hand and rubbed on paper to remove pencil marks or, in some cases, pen marks.”).
2. **Distractors List:** The identified entities to be excluded (e.g. *pencil*, *car tyre*, etc. in the case of Figure 1).
3. **Weighted Questions:** A set of 10 binary questions, divided into 5 *Positive Questions* (capturing the necessary visual features of the target) which will strengthen the visual signal of the correct image and 5 *Negative Questions* (specifically designed to rule out the visual features of the identified distractors) which will lower the visual signal of the wrong images.
4. **Importance Weights:** Inspired by [21], where the authors demonstrate that asking for a probability to the language model improves its generation variability, each question is associated with a generated weight $w \in [0.5, 1.0]$, representing its estimated discriminative power in the disambiguation process. This refinement enabled the production of a broader spectrum of questions, ensuring greater semantic variability compared to the simple version.

3.2. Phase 2: Visual Assessment

In the second phase, we verify whether candidate images satisfy a set of generated visual constraints. We employ **Qwen3-VL-8b**, a relatively small state-of-the-art Visual Question Answering (VQA) model, in a zero-shot setting. The model iterates through the set of $K = 10$ candidate images provided by the task. For each image, it answers all 10 generated questions. Crucially, we do not restrict the model to a binary text output; instead, we instruct it via prompt to answer exclusively “Yes” or “No” and access the underlying logits. We define the probability of the “Yes” token as $p(\text{Yes})$ and set $p(\text{No}) = 1 - p(\text{Yes})$. Although the model technically remains free to generate arbitrary sequences, in which case we would default to $p(\text{Yes}) = p(\text{No}) = 0.5$, this deviation was not observed in our experiments.

3.3. Phase 3: VQA Confidence Estimation

In order to determine a robust measure of the visual assessment above, we map the VQA output to a numerical score based on token probabilities. For each image I and each question q_j in the set of generated questions $Q(I)$, we query the VQA model and extract the probability of the first generated token. Let $p_{\text{YES}}^{(j)}$ be the probability associated with the “YES” tokens.

We define the local score s_j for the j -th question depending on its expected polarity (i.e., whether the image *should* or *should not* exhibit the feature):

$$s_j = \begin{cases} p_{\text{YES}}^{(j)} & \text{if } q_j \in Q(I)_{\text{positive}} \\ 1 - p_{\text{YES}}^{(j)} & \text{otherwise} \end{cases} \quad (1)$$

Finally, the global VQA score S_{VQA} for the image is computed as the geometric mean of the local scores. We explicitly chose the geometric mean (computed via log-space summation) over the arithmetic mean to penalise images that fail even a few critical questions (e.g., an image that looks like the target but fails a specific negative check against a co-hyponym). The score is calculated as follows, where $N = 10$ is the total number of questions:

$$S_{\text{VQA}} = \exp \left(\frac{1}{N} \sum_{j=1}^N \log(s_j) \right) \quad (2)$$

For the example in Figure 1, we show here a couple of questions with their answers and scores for each image:

1. **Image A, final score 0.03:**

- **Positive Question:** “L’oggetto è pensato per rimuovere segni da un foglio di carta?” (en: “Is the object designed to remove marks from a sheet of paper?”) → No (confidence 0.94 → score $1 - 0.94 = 0.06$)
- **Negative Question:** “L’oggetto è un grande anello nero con battistrada montato su una ruota?” (en: “Is the object a large black ring with treads mounted on a wheel?”) → Yes (confidence 0.97 → score $1 - 0.97 = 0.03$)

2. **Image B, final score 0.69:**

- **Positive Question:** “L’oggetto è pensato per rimuovere segni da un foglio di carta?” (en: “Is the object designed to remove marks from a sheet of paper?”) → Yes (confidence 0.99 → score 0.99)
- **Negative Question:** “L’oggetto è un grande anello nero con battistrada montato su una ruota?” (en: “Is the object a large black ring with treads mounted on a wheel?”) → No (confidence 0.77 → score 0.77)

3. **Image C, final score 0.47:**

- **Positive Question:** “L’oggetto è pensato per rimuovere segni da un foglio di carta?” (en: “Is the object designed to remove marks from a sheet of paper?”) → No (confidence 0.64 → score $1 - 0.64 = 0.36$)
- **Negative Question:** “L’oggetto è un grande anello nero con battistrada montato su una ruota?” (en: “Is the object a large black ring with treads mounted on a wheel?”) → No (confidence 0.89 → score 0.89)

3.4. Phase 4: Multimodal Semantic Matching

While the VQA module handles fine-grained details, we maintain a global perspective to capture the general “vibe” and semantic composition of the image. We use **CLIP** to compute the cosine similarity between the image embeddings and the text embeddings. To maximise robustness, we compute this similarity from two different linguistic views:

- $S_{Original}$: The similarity between the candidate image and the **Original Query** (the input sequence provided by the competition for the image I). This represents the noisy but grounded truth. As an example from Figure 1, CLIP produces the following scores with the original query: A 0.26, B 0.29, C 0.27, which implies a very poor similarity, even with the original image, but still a positive one.
- $S_{Synthetic}$: The similarity between the candidate image and the **Generated Holistic Description** produced in Phase 1. This represents a cleaner, richer, and syntactically well-formed description of the target concept, which usually contains different and more general words to describe the concept, even multi-faceted. For the example in Figure 1, CLIP produces the following scores with the synthetic description: A 0.15, B 0.39, C 0.33, which implies a better discrimination between the correct and wrong images.

3.5. Phase 5: Multi-Channel Fusion and Ranking

The final decision is produced by fusing the local verification score with the global semantic matching scores. The final score for each image can be computed as an unweighted sum of different combinations of the components, as determined empirically during the development phase (see Section 4):

$$Score_{Final} = S_{VQA} + S_{Original} + S_{Synthetic} \quad (3)$$

The candidate images are sorted by $Score_{Final}$ in descending order, and the top-ranked image is selected as the answer.

4. Results and Discussion

In this section, we present the experimental evaluation of the proposed architecture. We first detail the setup and the tuning process performed on the development set, then report the official results achieved in the EVWSD-ITA task, discuss an ablation study concerning the impact of structured query parsing, and finally present an error analysis over a borderline example.

4.1. Experimental Setup

Our pipeline relies on the following components:

- **Reasoning Engine:** We utilised **GPT-5.1**¹ for the semantic analysis and question generation phases.
- **VQA Model:** We employed **Qwen3-VL-8b**², a relatively small state-of-the-art vision-language model, to perform the zero-shot verification of the generated questions.

¹Accessed via OpenAI API in December 2025

²<https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct>

- **Vision-Language Encoder:** For global semantic matching, we used CLIP³ to compute cosine similarities between image and text embeddings.

The evaluation metrics adopted for the task are **HIT@1** (accuracy of the top-1 retrieved image) and **MRR** (Mean Reciprocal Rank), as detailed in the shared task paper.

4.2. Development and Tuning

Since we do not perform any training and an official validation set was not provided, we employed the entire training set as a development set to calibrate the system components and select the optimal configuration. We performed an extensive grid search to determine the best strategies for question selection and score fusion.

Question Selection. We investigated the impact of the number and type of questions (k) on the disambiguation performance. We compared using only Positive questions (varying $k \in [1, 5]$) against using the full battery of Positive and Negative questions. Experiments showed that Negative questions are essential for filtering hard negatives (co-hyponyms), and the best performance was achieved using the maximum set size (5 Positive + 5 Negative).

Fusion Strategy. We compared a Weighted Sum approach against a Max-Pooling strategy. We tested various coefficients for the linear combination:

$$Score = w_1 \cdot S_{VQA} + w_2 \cdot S_{Original} + w_3 \cdot S_{Synthetic} \quad (4)$$

Experimental results indicated that the system is robust enough that complex weighting schemes are unnecessary. The most stable configuration was an **unweighted sum** ($w_1 = w_2 = w_3 = 1$), which treats the evidence from the original query, the synthetic description, and the visual verification as equally important signals.

4.3. Main Results

We submitted three runs to the EVWSD-ITA challenge. The first two runs represent our primary generation pipeline, while the third explores an alternative structured approach, which is detailed in Section 4.4.

- **Run 1 (VQA+Or):** This run combines the VQA verification score with the global CLIP similarity of the original query. The final score is computed as:

$$Score_{Run1} = S_{VQA} + S_{Original} \quad (5)$$

- **Run 2 (VQA+Or+Desc):** This is our complete pipeline. It adds the global CLIP similarity of the *Synthetic Holistic Description* generated by the LLM. The final score corresponds to the full fusion formula:

$$Score_{Run2} = S_{VQA} + S_{Original} + S_{Synthetic} \quad (6)$$

Discussion. Table 1 summarises the performance of our Zero-Shot architecture on the EVWSD-ITA test set. The results provide strong empirical evidence answering our initial research questions. The standard **CLIP Baseline**, which relies solely on the cosine similarity between the candidate images and the raw input triple, achieves a HIT@1 of 45.05%. While this performs significantly better than random chance (10%), it highlights the inherent difficulty of the task: standard dual-encoders struggle to distinguish the target from Hard Negatives (co-hyponyms) that share a similar embedding space. This confirms the “bag-of-words” limitation discussed in Section 2. A dramatic improvement is observed when moving to **Run 0** (VQA only), which uses the aggregated confidence scores of the generated questions without any global CLIP signal. This configuration achieves 65.32% HIT@1, surpassing the baseline by over 20 percentage points. This result provides a definitive answer to **RQ1** and **RQ2**:

³<https://huggingface.co/sentence-transformers/clip-ViT-B-32-multilingual-v1>

Table 1

Official results on the EVWSD-ITA test set. **Run 2** represents our best-performing configuration. *Or* stands for the CLIP score for each image against the original input sequence and *Desc* for the CLIP score against the description generated by the LLM, as detailed in the previous Section.

Run ID	HIT@1	MRR
CLIP (Baseline)	45.05%	62.44%
Run 0 (VQA)	65.32%	76.82%
Run 1 (VQA+Or)	69.37%	81.00%
Run 2 (VQA+Or+Desc)	71.17%	81.82%
Run 3 (2stepVQA+Or+Desc)	66.67%	77.70%

modelling EVWSD as a *Visual Constraint Verification* task is far more effective than global embedding matching for fine-grained disambiguation. The VQA model, guided by LLM-generated positive and negative constraints, successfully filters out hard negatives that visually resemble the target but lack specific distinguishing features (e.g., specific shapes or usage contexts). Comparing **Run 0** with **Run 1** demonstrates the value of the ensemble approach (**RQ4**). Adding the original CLIP score ($S_{Original}$) to the VQA score improves performance from 65.32% to 69.37%. This suggests that while local feature verification (VQA) is crucial for precision, the global semantic signal from CLIP helps maintain a holistic view, acting as a safeguard when the generated questions might be too specific or ambiguous. The two modules are complementary: VQA handles the fine-grained details, while CLIP ensures global coherence. Finally, comparing **Run 1** and **Run 2** allows us to address **RQ3**. The inclusion of the $S_{Synthetic}$ score in Run 2 yields a further performance improvement, reaching the best result of **71.17%** HIT@1 and **81.82%** MRR. This improvement of approximately 1.8% confirms that the raw query provided by the dataset is often too noisy or fragmented for optimal global matching. The LLM-generated description acts as a “semantic bridge”, providing a well-formed text that aligns better with the pre-training distribution of the visual encoder, giving a final boost to the accuracy of our system.

4.4. Ablation Study: The Cost of Structure

To investigate whether explicit linguistic structuring benefits the reasoning process, we submitted a third run based on a **Two-Step Generation** policy. In this configuration, we forced the LLM to first parse the noisy input string into a formal structure (separating Lemma, Hypernym, and Gloss) and then generate the visual questions based solely on this intermediate representation. The final score aggregation remains identical to Run 2 (Equation 6).

The results for **Run 3**, reported in Table 1, offer a crucial insight into prompt engineering for VWSD. With a HIT@1 of **66.67%**, this structured approach significantly outperforms the CLIP Baseline (+21.62%), confirming that the core VQA-based verification strategy is sound regardless of the generation nuance. However, compared to our best configuration (**Run 2**), Run 3 suffers a performance drop of **4.5 percentage points** in HIT@1 and over 4 points in MRR.

This substantial gap refutes the intuition that cleaner, structured inputs necessarily lead to better questions in this context. We attribute this failure to *error propagation*: the rigid “Parse → Generate” pipeline creates an information bottleneck. If the LLM introduces even a minor hallucination or misinterpretation during the initial parsing (e.g., mis-assigning a polysemous word from the gloss to the hypernym field), this error becomes hard-coded in the intermediate structure, irretrievably compromising the subsequent question generation. In contrast, the **Direct Generation** approach (Runs 1 and 2) allows the LLM to leverage its full attention mechanism over the raw, noisy context. This enables the model to resolve ambiguities *implicitly* and dynamically during the generation of questions, proving that for high-capacity models like GPT-5.1, an end-to-end reasoning process is more robust than a formally decomposed pipeline.

4.5. Error Analysis



Figure 3: Top-5 retrieval results for the query “sella borsa bisaccia”, showing a preference for a mounted usage context (17 . jpg) over the gold product image (1216 . jpg).

Table 2

Comparison of VQA responses between the predicted image (17 . jpg) and the gold image (1216 . jpg) for the query “sella borsa bisaccia”. The positive (first 5) questions favour the mounted context, while the negative (last 5) questions are not discriminative enough. The Score for each question is computed as $p(YES)$ for the positive ones and $1 - p(YES)$ for the negative ones, as detailed in Section 3, Phase 3.

Question	17 . jpg (Predicted)			1216 . jpg (Gold)		
	Answer	$p(YES)$	Score	Answer	$p(YES)$	Score
<i>Si vedono due borse morbide fissate ai lati di una sella?</i>	YES	0.6225	0.6225	NO	0.0180	0.0180
<i>Le borse sono collegate tra loro e poggiano simmetricamente sul dorso del mezzo?</i>	YES	0.7311	0.7311	NO	0.0373	0.0373
<i>Le borse sembrano progettate per contenere oggetti durante uno spostamento?</i>	YES	1.0000	1.0000	YES	0.9998	0.9998
<i>Il materiale delle borse appare robusto, come cuoio o tessuto spesso?</i>	YES	0.7773	0.7773	YES	0.9968	0.9968
<i>Le borse sono posizionate dietro o accanto al sedile principale del cavallo o moto?</i>	NO	0.1480	0.1480	NO	0.2689	0.2689
<i>Si vede solo un sedile rigido senza borse laterali agganciate?</i>	NO	0.0000	1.0000	NO	0.0000	1.0000
<i>L’oggetto è una borsa singola portata a spalla da una persona?</i>	NO	0.0000	1.0000	NO	0.0000	1.0000
<i>L’oggetto è uno zaino con spallacci indossato sulla schiena?</i>	NO	0.0000	1.0000	NO	0.0000	1.0000
<i>Il contenitore è un bauletto rigido montato centralmente sul portapacchi?</i>	NO	0.0000	1.0000	NO	0.0000	1.0000
<i>La borsa è appesa al manubrio senza appoggiarsi ai lati della sella?</i>	NO	0.0180	0.9820	NO	0.0000	1.0000

To illustrate the failure modes of the proposed VQA-based scoring strategy, we analyse a representative example in which the predicted top-ranked image differs from the gold annotation. The following discussion focuses on two recurring phenomena observed across multiple errors: the effect of ambiguous images providing partial but question-aligned visual evidence, and the impact of negative questions that are logically correct yet insufficiently discriminative. The example reported in Figure 3 and Table 2 serves as a concrete case study to highlight how these factors interact and influence the final ranking.

In the analysed case, the system is strongly influenced by the presence of partial visual evidence that directly matches the formulation of the VQA questions. The first two questions explicitly assume the visibility of saddlebags already mounted on a saddle, implicitly requiring the presence of the vehicle context. Image 17 . jpg satisfies these assumptions by depicting a bicycle with lateral bags mounted, and consequently receives affirmative answers with medium-to-high confidence. In contrast, the gold image 1216 . jpg, while correctly representing a saddle bag as an object, is photographed in isolation

and does not include the saddle or the vehicle. As a result, the same questions yield near-zero scores, penalising the gold image despite its higher semantic relevance to the query. This behaviour indicates that the system prioritises local visual alignment with the questions over the correct identification of the target object class.

In addition, the example highlights the role of negative questions that are correct but poorly aligned with the discrimination task. The last five questions in Table 2 exclude clearly distinct categories such as backpacks, shoulder bags, rigid top cases, or handlebar bags. Both images correctly receive negative answers with near-maximal confidence, leading to similar score contributions. While these questions are valid from a logical standpoint, they do not help distinguish between semantically close candidates, such as mounted lateral bags and saddle bags depicted as standalone products. Their contribution to the aggregated score is therefore inflationary rather than informative, amplifying the advantage already accumulated by the image favoured by context-dependent questions. This behaviour suggests that negative questions should be carefully selected or weighted according to their actual discriminative power, in order to avoid trivial exclusions dominating the decision process.

5. Conclusions

In this paper, we presented the UniTor system for the EVWSD-ITA task at EVALITA 2026. We proposed a fully Zero-Shot, training-free architecture that redefines Visual Word Sense Disambiguation as a process of *Visual Constraint Verification*. By leveraging the reasoning capabilities of Large Language Models to guide the discriminative power of Visual Question Answering models, our approach moves beyond the limitations of standard global embeddings.

Our experimental results allow us to draw three main conclusions regarding the initial research questions:

1. **Verification outperforms Matching:** The substantial performance gap between our VQA-based runs and the CLIP baseline confirms that decomposing a query into fine-grained Positive and Negative questions is essential for distinguishing the target from hard negatives, such as co-hyponyms.
2. **Context Reconstruction is beneficial:** Augmenting the noisy input with an LLM-generated holistic description provides a cleaner semantic signal, offering a robust bridge between the linguistic and visual modalities (**Run 2**).
3. **Implicit Reasoning is robust:** The ablation study on structured parsing (**Run 3**) revealed that enforcing a rigid “Parse → Generate” pipeline introduces information bottlenecks. Direct generation proves to be superior, suggesting that state-of-the-art LLMs manage ambiguity better when allowed to attend to the raw context implicitly.

Future work will focus on the efficiency of the pipeline and a thorough investigation of the selection of negative questions. While the current architecture achieves high accuracy, it relies on large, API-based models. We plan to investigate whether smaller, open-source Language Models can be fine-tuned to generate high-quality visual constraints, making the approach more sustainable and deployable in resource-constrained environments.

Acknowledgments

We acknowledge financial support from the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU and support from Project ECS 0000024 Rome Technopole - CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, funded by the European Union - NextGenerationEU.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

- [1] T. Baltrušaitis, C. Ahuja, L.-P. Morency, Multimodal machine learning: A survey and taxonomy, *IEEE transactions on pattern analysis and machine intelligence* 41 (2018) 423–443.
- [2] A. Mogadala, M. Kalimuthu, D. Klakow, Trends in integration of vision and language research: A survey of tasks, datasets, and methods, *J. Artif. Intell. Res.* 71 (2021) 1183–1317. URL: <https://doi.org/10.1613/jair.1.11688>. doi:10.1613/JAIR.1.11688.
- [3] S. Harnad, The symbol grounding problem, *Physica D: Nonlinear Phenomena* 42 (1990) 335–346.
- [4] A. Raganato, I. Calixto, A. Ushio, J. Camacho-Collados, M. T. Pilehvar, SemEval-2023 task 1: Visual word sense disambiguation, in: A. K. Ojha, A. S. Doğruöz, G. Da San Martino, H. Tayyar Madabushi, R. Kumar, E. Sartori (Eds.), *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Association for Computational Linguistics, Toronto, Canada, 2023, pp. 2227–2234. URL: <https://aclanthology.org/2023.semeval-1.308/>. doi:10.18653/v1/2023.semeval-1.308.
- [5] F. Cutugno, A. Miaschi, A. Palmero Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, EVALITA 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [6] A. K. Ojha, A. S. Doğruöz, G. Da San Martino, H. Tayyar Madabushi, R. Kumar, E. Sartori (Eds.), *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, Association for Computational Linguistics, Toronto, Canada, 2023. URL: <https://aclanthology.org/2023.semeval-1.0/>.
- [7] R. Navigli, S. P. Ponzetto, Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, *Artificial intelligence* 193 (2012) 217–250.
- [8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, *CoRR abs/2103.00020* (2021). URL: <https://arxiv.org/abs/2103.00020>. arXiv:2103.00020.
- [9] X. Zhang, T. Zhen, J. Zhang, Y. Wang, S. Liu, Srcb at semeval-2023 task 1: Prompt based and cross-modal retrieval enhanced visual word sense disambiguation, in: *Proceedings of the 17th international workshop on semantic evaluation (SemEval-2023)*, 2023, pp. 439–446.
- [10] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, E. Chen, A survey on multimodal large language models, 2024. URL: <https://arxiv.org/abs/2306.13549>. arXiv:2306.13549.
- [11] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, J. Lin, Qwen2.5-vl technical report, 2025. URL: <https://arxiv.org/abs/2502.13923>. arXiv:2502.13923.
- [12] C. D. Hromei, A. Scaiella, D. Croce, R. Basili, Grounded semantic role labelling from synthetic multimodal data for situated robot commands, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Suzhou, China, 2025, pp. 23758–23781. URL: <https://aclanthology.org/2025.emnlp-main.1212/>. doi:10.18653/v1/2025.emnlp-main.1212.
- [13] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin,

- X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, K. Zhu, Qwen3-vl technical report, 2025. URL: <https://arxiv.org/abs/2511.21631>. arXiv:2511.21631.
- [14] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, J. Zou, When and why vision-language models behave like bags-of-words, and what to do about it?, arXiv preprint arXiv:2210.01936 (2022).
 - [15] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems* 35 (2022) 24824–24837.
 - [16] S. Shen, L. H. Li, H. Tan, M. Bansal, A. Rohrbach, K.-W. Chang, Z. Yao, K. Keutzer, How much can clip benefit vision-and-language tasks?, arXiv preprint arXiv:2107.06383 (2021).
 - [17] A. Shtedritski, C. Rupprecht, A. Vedaldi, What does clip know about a red circle? visual prompt engineering for vlms, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11987–11997.
 - [18] C. Cuttano, G. Rosi, G. Trivigno, G. Averta, What does clip know about peeling a banana?, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 2238–2247.
 - [19] R. Ramos, V. Stojnić, G. Kordopatis-Zilos, Y. Nakashima, G. Tolas, N. Garcia, Processing and acquisition traces in visual encoders: What does clip know about your camera?, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 17056–17066.
 - [20] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, D. Parikh, Vqa: Visual question answering, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2425–2433.
 - [21] J. Zhang, S. Yu, D. Chong, A. Sicilia, M. R. Tomz, C. D. Manning, W. Shi, Verbalized sampling: How to mitigate mode collapse and unlock llm diversity, 2025. URL: <https://arxiv.org/abs/2510.01171>. arXiv:2510.01171.

A. Appendix

In this Appendix, we report the prompts used in our pipeline, which are the prompt for GPT-5.1 for generating the questions (A.1), the VQA prompt for answering the generated questions (A.2) and the two prompts used in the Run3 of our ablation study (A.3).

A.1. Question Generator Prompt

This prompt instructs the GPT-5.1 model to adopt the persona of a computational linguist specialised in visual and lexical semantics. The primary objective is to generate a structured JSON output that profiles the semantic content of a given text segment (k words) for the purpose of assessing image-text consistency. The procedure operates in two phases in the same answer, conditioning the question generation step on the identified sense. First, the model performs a semantic analysis to identify the concrete visual target (synset) and its hypernym, while simultaneously detecting potential semantic ambiguities, including co-hyponyms and polysemous alternative senses. Second, the model constructs a JSON object containing the identified target, a functional description, and a set of ten weighted binary questions (five confirming, five discriminating). These questions are strictly grounded in visual properties (shape, texture, colour) to validate the presence of the concept. Each question is assigned a weight ($w \in [0.5, 1.0]$) indicating either its representativeness of the target or its power to discriminate the target from semantic distractors.

Sei un linguista computazionale esperto di semantica visiva e lessicale.

Il tuo obiettivo è creare un JSON che rappresenti la conoscenza semantica di un contesto testuale contenente k parole e che serva per valutare se un'immagine rappresenta correttamente il concetto espresso dal contesto.

FASE 1 – ANALISI SEMANTICA

1. Considera il contesto testuale come un frammento estratto da una rete semantica (tipo BabelNet o WordNet).
2. Identifica (ma non con questo ordine):
 - l'OGGETTO VISIVO CONCRETO (synset, target visivo);
 - la CATEGORIA GENERALE (iperonimo);
 - la sua descrizione (solitamente una parola estratta dalla sua glossa).
3. Elenca 3-6 concetti affini o potenzialmente confondibili (co-iponimi o sensi alternativi). Questi rappresentano le "distrazioni semantiche".

FASE 2 – GENERAZIONE DEL JSON

Genera un JSON nel formato seguente:

```
{
  "query": "<contesto di k parole>",
  "identified_target": "<synset, l'oggetto visivo concreto>",
  "description": "<breve spiegazione visiva e funzionale del target identificato>",
  "questions_yes": [
    {"testo": "<domanda>", "w": <valore tra 0.5 e 1.0>},
    {"testo": "<domanda>", "w": <valore tra 0.5 e 1.0>},
    ...
  ],
  "hypothesized_co_hyponyms": [
    "<termine1>",
    "<termine2>",
    "<termine3>",
    ...
  ],
  "senses_to_exclude": [
    "<senso alternativo 1>",
    "<senso alternativo 2>",
    "<senso alternativo 3>",
    ...
  ],
  "questions_no": [
    {"testo": "<domanda>", "w": <valore tra 0.5 e 1.0>},
    {"testo": "<domanda>", "w": <valore tra 0.5 e 1.0>},
    ...
  ]
}
```

ISTRUZIONI PER LE DOMANDE

- Scrivi tutto in ITALIANO corretto.
- Domande concise (al più 15 parole), rispondibili con "si" o "no".
- Basate su proprietà visive concrete (forma, materiale, colore, posizione, funzione, ecc. Tutto ciò che distingue il concetto target da qualunque altro senso possibile).
- Genera esattamente 5 domande "si" e 5 domande "no".
- Ogni domanda ha un campo `w`:
 - Per le **questions_yes**: indica quanto fortemente la domanda coglie l'essenza visiva del target identificato.
 - Per le **questions_no**: indica quanto fortemente la domanda è discriminante nel distinguere il target identificato dai sensi alternativi.

LINEA GUIDA SEMANTICA

- Le domande con `w` più alto devono essere le più rappresentative o discriminanti.
- Es: per le questions_yes, quella con `w`=1.0 deve esprimere la caratteristica più distintiva del concetto target.

- Es: per le questions_no, quella con `w`=1.0 deve cogliere la differenza chiave che fa capire che NON si tratta del concetto target.
- Se utile, immagina di addestrare un modello visivo-linguistico: le domande con `w` alto sono le più informative per capire il concetto.

ESEMPIO

Input: "attrezzo gomma da cancellare atto"

Output:

```
{
  "query": "attrezzo gomma da cancellare atto",
  "identified_target": "Gomma da cancellare",
  "description": "Piccolo blocco di materiale gommoso o vinilico utilizzato per rimuovere tratti di matita o inchiostro dalla carta tramite abrasione.",
  "questions_yes": [
    {
      "testo": "L'oggetto sembra progettato per essere sfregato contro una superficie cartacea?",
      "w": 1.0
    },
    {
      "testo": "L'oggetto si presenta come un piccolo blocco solido, spesso a forma di parallelepipedo?",
      "w": 0.9
    },
    {
      "testo": "L'oggetto ha una texture liscia e opaca, tipica della gomma o del vinile?",
      "w": 0.8
    },
    {
      "testo": "L'oggetto è tipicamente bianco, rosa o bicolore rosso e blu?",
      "w": 0.7
    },
    {
      "testo": "L'oggetto è contestualizzato vicino a strumenti di scrittura come matite o fogli?",
      "w": 0.6
    }
  ],
  "hypothesized_co_hyponyms": [
    "Temperamatite",
    "Bianchetto liquido",
    "Correttore a nastro",
    "Righello"
  ],
  "senses_to_exclude": [
    "Pneumatico (gomma dell'auto)",
    "Gomma da masticare (chewing gum)",
    "Elastico (anello di gomma)",
    "Caucciù (materiale grezzo)"
  ],
  "questions_no": [
    {
      "testo": "L'oggetto è un grande anello nero con battistrada montato su un cerchione?",
      "w": 1.0
    },
    {
      "testo": "L'oggetto è un piccolo confetto o lastrina confezionata per uso alimentare?",
      "w": 0.9
    },
    {
      "testo": "L'oggetto è un sottile anello flessibile usato per tenere insieme altri oggetti?",
      "w": 0.8
    },
    {
      "testo": "L'oggetto è un flaconcino contenente un liquido bianco con un applicatore?",
      "w": 0.7
    },
    {
      "testo": "L'oggetto presenta lame metalliche visibili per affilare il legno?",
      "w": 0.6
    }
  ]
}
```

```
]
}
```

Ora elabora il seguente concetto:
\$QUERY\$

Genera SOLO il JSON richiesto, senza testo aggiuntivo o spiegazioni.

A.2. Prompt for VQA

The VQA (Qwen3-VL-8b) inference prompt is a simple request to answer the question with just Yes or No, in order to verify the visual features and extract the associated probabilities as scores.

Stai svolgendo un compito di Visual Question Answering (VQA).

Ti fornisco un'immagine e una domanda. La domanda è stata generata a partire da tre indizi lessicali che definiscono un concetto target (lemma del concetto, lemma del suo iperonimo e un termine di contesto estratto dalla glossa). L'ordine dei tre indizi è arbitrario.

Indizi: \$INPUT QUERY\$

Istruzioni:

- Rispondi basandoti esclusivamente su ciò che è visibile nell'immagine.
- Se l'evidenza visiva è assente o ambigua, rispondi NO.
- Produci come output una sola parola: YES oppure NO.
- Scrivi YES/NO come primo token e non aggiungere altro testo.

Domanda: \$GENERATED QUESTIONS\$

A.3. Prompt for Run3: Structured Parsing and Question Generation

In this section, we report the prompts used for the third run of our submission, i.e., the one where we first ask GPT-5.1 to parse the input concept into a structured object, and then to generate the list of questions.

The prompt for parsing the concept into a structured form is the following:

Sei un esperto semantico specializzato in analisi lessicale e ontologie (p.e., BabelNet).

Il tuo compito è analizzare una stringa di testo che contiene tre componenti semantici mescolati in ordine casuale e senza separatori espliciti. Devi segmentare la stringa e ricostruire la struttura logica originale.

I tre componenti nascosti nella stringa sono:

1. "lemma_concetto": Il concetto specifico (Target). Può essere composto da una o più parole (es. "gomma da cancellare").
2. "lemma_iperonimo": La categoria generale a cui appartiene il concetto. Può essere composto da una o più parole (es. "attrezzo"). Relazione: [Concetto] È UN [Iperonimo].
3. "parola_glossa": Una singola parola estratta dalla definizione del dizionario (glossa) del concetto.

ISTRUZIONI DI RAGIONAMENTO:

- Identifica la relazione gerarchica: cerca la coppia di termini dove uno è il "padre" (iperonimo) e l'altro è il "figlio" (concetto).
- Riconosci le espressioni polirematiche: in italiano molti concetti sono formati da più parole (es. "carta di credito", "mulino ad acqua"). Non spezzarle.
- La parola rimanente, che non è né il concetto né l'iperonimo, è la parola della glossa.

ESEMPI RISOLTI (Few-Shot Learning):

Esempio 1:

Input: "strumento musicale tasti pianoforte"

Analisi: "pianoforte" è un tipo di "strumento musicale". "tasti" è una parola che descrive il pianoforte.

Output JSON:

```
{
  "lemma_concetto": "pianoforte",
  "lemma_iperonimo": "strumento musicale",
  "parola_glossa": "tasti"
}
```

Esempio 2 (Ordine diverso, concetto multi-parola):

Input: "felino gatto domestico mammifero"

Analisi: Qui c'è ambiguità, ma "gatto domestico" è il concetto più specifico. "mammifero" è la categoria più alta, ma spesso l'iperonimo diretto è "felino". Se la stringa fosse "gatto domestico felino mammifero", "felino" è l'iperonimo diretto. Assumiamo: concetto="gatto domestico", iperonimo="felino", glossa="mammifero".

Output JSON:

```
{
  "lemma_concetto": "gatto domestico",
  "lemma_iperonimo": "felino",
  "parola_glossa": "mammifero"
}
```

Esempio 3 (Iperonimo multi-parola, ordine mescolato):

Input: "pagamento carta di credito tessera magnetica"

Analisi: "carta di credito" è il concetto. "tessera magnetica" è la categoria (iperonimo). "pagamento" è l'uso (glossa).

Output JSON:

```
{
  "lemma_concetto": "carta di credito",
  "lemma_iperonimo": "tessera magnetica",
  "parola_glossa": "pagamento"
}
```

Esempio 4 (Il tuo caso specifico):

Input: "attrezzo gomma da cancellare atto"

Analisi: "gomma da cancellare" è l'oggetto specifico. "attrezzo" è la categoria generale. "atto" è la parola della descrizione.

Output JSON:

```
{
  "lemma_concetto": "gomma da cancellare",
  "lemma_iperonimo": "attrezzo",
  "parola_glossa": "atto"
}
```

TASK CORRENTE:

Analizza la stringa seguente e restituisci SOLO il JSON formattato.

Input: \$INPUT CONCEPT\$

The prompt to generate the questions from the structured concept is the following:

Sei un esperto di Vision-Language e Prompt Engineering per la challenge EVALITA VWSD.

Il tuo input è un oggetto JSON che definisce un concetto visivo preciso, già disambiguato.

Input JSON:

```
{
  "lemma_concetto": "<Target Visivo>",
  "lemma_iperonimo": "<Categoria/Padre>",
  "parola_glossa": "<Contesto/Disambiguatore>"
}
```

OBIETTIVO:

Generare una batteria di domande SI/NO in **italiano** da sottoporre a un modello CLIP/VLM per capire se un'immagine candidata rappresenta **ESATTAMENTE** il "lemma_concetto".

STRATEGIA AVANZATA:

Il modello visivo deve distinguere il Target dai "Hard Negatives" (immagini molto simili ma sbagliate).

1. Se il Target è polisemico (ha più significati), usa la "parola_glossa" per capire quale senso è

quello giusto ed escludere gli altri.

- Se il Target ha "fratelli" nella stessa categoria (co-iponimi), devi generare domande che evidenzino le differenze sottili.
 - Esempio: Target="Ghepardo", Iperonimo="Felino".
 - Hard Negative: "Leopardo".
 - Domanda NO efficace: "Le macchie sul mantello sono a forma di rosetta con il centro vuoto?" (Vero per Leopardo, Falso per Ghepardo).

REGIONE DI GENERAZIONE:

- **Solo Visuale:**** Le domande devono riguardare forme, colori, texture, layout, azioni visibili. Niente concetti astratti.
- **Niente "Target":** NON usare la parola contenuta in "lemma_concetto" nel testo delle domande. Descrivilo.
- **Domande SI (Positive):** Devono essere VERE solo per il Target.
- **Domande NO (Trappole):** Devono essere FALSE per il Target, ma VERE per i suoi simili (co-iponimi) o per i suoi altri sensi. Devono servire a scartare le immagini "quasi giuste".
- **Ogni domanda ha un campo "w":**
 - Per le ****questions_yes****: indica quanto fortemente la domanda coglie l'essenza visiva del target identificato.
 - Per le ****questions_no****: indica quanto fortemente la domanda è discriminante nel distinguere il target identificato dai sensi alternativi.

OUTPUT FORMAT (JSON):

Restituisci solo il JSON seguente.

```
{
  "description": "Breve descrizione del concetto che hai ricevuto in input, con focus particolare sul lemma target e sul suo iperonimo",
  "questions_yes": [
    {"<Domanda 1: Focus su attributo univoco del target>", "w": <score per la domanda 1>},
    {"<Domanda 2: Focus su forma/struttura>", "w": <score per la domanda 2>},
    {"<Domanda 3: Focus su contesto tipico>", "w": <score per la domanda 3>},
    {"<Domanda 4: Focus su dettagli specifici>", "w": <score per la domanda 4>},
    {"<Domanda 5: Conferma visiva finale>", "w": <score per la domanda 5>}
  ],
  "hypothesized_co_hyponyms": [
    "<termine1>",
    "<termine2>",
    "<termine3>",
    ...
  ],
  "senses_to_exclude": [
    "<sensu alternativo 1>",
    "<sensu alternativo 2>",
    "<sensu alternativo 3>",
    ...
  ],
  "questions_no": [
    {"<Domanda 1: Vera per un co-iponimo specifico, falsa per il target>", "w": <score per la domanda 1>},
    {"<Domanda 2: Vera per un altro senso della parola (se esiste), falsa per questo>", "w": <score per la domanda 2>},
    {"<Domanda 3: Vera per l'iperonimo in generale ma falsa per questo specifico sottotipo>", "w": <score per la domanda 3>},
    {"<Domanda 4: Trappola visiva su dettaglio errato>", "w": <score per la domanda 4>},
    {"<Domanda 5: Altra trappola per co-iponimo>", "w": <score per la domanda 5>}
  ]
}
```

ESEMPIO

Input:

```
{
  "lemma_concetto": "gomma da cancellare",
  "lemma_iperonimo": "attrezzo",
  "parola_glossa": "atto"
}
```

Output:

```
{
  "description": "Piccolo blocco di materiale gommoso o vinilico utilizzato per rimuovere tratti di matita o inchiostro dalla carta tramite abrasione.",
  "questions_yes": [
```

```

    {
      "testo": "L'oggetto sembra progettato per essere sfregato contro una superficie cartacea?",
      "w": 1.0
    },
    {
      "testo": "L'oggetto si presenta come un piccolo blocco solido, spesso a forma di
        parallelepipedo?",
      "w": 0.9
    },
    {
      "testo": "L'oggetto ha una texture liscia e opaca, tipica della gomma o del vinile?",
      "w": 0.8
    },
    {
      "testo": "L'oggetto è tipicamente bianco, rosa o bicolore rosso e blu?",
      "w": 0.7
    },
    {
      "testo": "L'oggetto è contestualizzato vicino a strumenti di scrittura come matite o fogli?",
      "w": 0.6
    }
  ],
  "hypothesized_co_hyponyms": [
    "Temperamatite",
    "Bianchetto liquido",
    "Correttore a nastro",
    "Righello"
  ],
  "senses_to_exclude": [
    "Pneumatico (gomma dell'auto)",
    "Gomma da masticare (chewing gum)",
    "Elastico (anello di gomma)",
    "Caucciù (materiale grezzo)"
  ],
  "questions_no": [
    {
      "testo": "L'oggetto è un grande anello nero con battistrada montato su un cerchione?",
      "w": 1.0
    },
    {
      "testo": "L'oggetto è un piccolo confetto o lastrina confezionata per uso alimentare?",
      "w": 0.9
    },
    {
      "testo": "L'oggetto è un sottile anello flessibile usato per tenere insieme altri oggetti?",
      "w": 0.8
    },
    {
      "testo": "L'oggetto è un flaconcino contenente un liquido bianco con un applicatore?",
      "w": 0.7
    },
    {
      "testo": "L'oggetto presenta lame metalliche visibili per affilare il legno?",
      "w": 0.6
    }
  ]
}

```

INPUT: \$PARSED CONTEXT\$

ATE-IT at EVALITA 2026: Overview of the Automatic Term Extraction Italian Testbed Task

Nicola Cirillo^{1,*}, Giorgio Maria Di Nunzio^{2,†} and Federica Vezzani^{3,†}

¹Department of Political and Communication Sciences, University of Salerno, 132 Via Giovanni Paolo II, Fisciano (SA), 84084, Italy

²Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

³Department of Linguistic and Literary Studies, University of Padova, Via Elisabetta Vendramini, 13 35137 Padova, Italy

Abstract

This paper presents an overview of the Automatic Term Extraction Italian Testbed (ATE-IT) shared task, organised within the EVALITA 2026 evaluation campaign. The task addresses the scarcity of benchmarks for Italian Automatic Term Extraction (ATE) by proposing a challenge focused on the domain of municipal waste management. Participants were invited to tackle two subtasks: (A) *Term Extraction*, aiming to identify domain-specific terms in institutional texts, and (B) *Term Variants Clustering*, focusing on grouping morphological and semantic term variants. Nine teams participated, submitting a total of 13 runs. The comparative analysis reveals that fine-tuned Transformer architectures generally outperform naive zero-shot Large Language Model (LLM) prompting, while hybrid approaches appear most effective for semantic clustering.

Keywords

Automatic Term Extraction, Terminology, Italian NLP, Shared Task, EVALITA 2026

1. Introduction

Automatic Term Extraction (ATE) is a foundational task in NLP and terminology work [1, 2]. Its goal is to identify domain-specific terms that designate key concepts within a specialised field of knowledge.

Although it shares some similarities with Named Entity Recognition (NER) [3], ATE differs from it. NER involves identifying and classifying mentions of named entities in running text (e.g., people, organisations, places, dates, etc.). Its focus is usually on proper names or unique references that have a clear instance-level referent, and the output is often linked to knowledge bases (e.g., “Barack Obama →Person”, “Google →Organization”). ATE, by contrast, aims to extract domain-specific terms from a corpus. This means identifying both multi-word and single-word terms that are relevant to a specialised field of knowledge (e.g., “informed consent”, “cryptic species”, “blockchain consensus algorithm”). The terms extracted through ATE serve as essential building blocks for downstream tasks such as information retrieval, machine translation, ontology construction, knowledge graph enrichment, and domain adaptation of large language models (LLMs).

In this paper, we describe the Automatic Term Extraction Italian Testbed (ATE-IT) shared task, organised in the context of EVALITA 2026, the 9th evaluation campaign of NLP and speech tools for Italian [4]. ATE-IT is the first large-scale evaluation campaign on Italian ATE, centred on a clearly defined real-world scenario: terminology extraction from institutional texts in the domain of waste management. This domain presents a wide variety of derived terms (e.g., “ecodizionario”, “biodigestore”), synonyms (e.g., “indifferenziato” and “secco residuo”), abbreviations (“TARI”, “RAEE”), and multiword expressions (“mastello contenitore”, “raccolta porta a porta”), making it an important testbed for assessing the robustness of different approaches.

All datasets, evaluation scripts, and baseline code are publicly available at the task repository.¹

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ nicirillo@unisa.it (N. Cirillo); giorgiomaria.dinunzio@unipd.it (G. M. D. Nunzio); federica.vezzani@unipd.it (F. Vezzani)

id 0000-0002-2107-1313 (N. Cirillo); 0000-0001-7116-9338 (G. M. D. Nunzio); 0000-0003-2240-6127 (F. Vezzani)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://github.com/nicolaCirillo/ate-it>

The remainder of this paper is organized as follows: Section 2 and Section 3 discuss the related work and the motivation behind the task. Sections 4 through 7 detail the experimental setup, including the task definition, dataset construction, evaluation measures, and the baseline system. Section 8 introduces the participating systems, while Section 9 and Section 10 present the official results and a discussion of the key findings. Finally, Section 11 concludes the paper.

2. Related Work

Automatic Term Extraction (ATE) has evolved significantly over the last few decades, transitioning from rule-based and statistical pipelines to deep learning and, most recently, prompt-based approaches. Traditional ATE systems generally follow a three-step pipeline: candidate extraction, feature calculation, and probability estimation [5, 6]. Candidate extraction typically relies on linguistic filters [7, 8], while feature calculation and probability estimation often exploit statistical measures [9, 10]. With the advent of deep learning, ATE has increasingly been framed as a sequence labeling task, bypassing the three-step pipeline. Early neural approaches utilized LSTMs [11], but the state-of-the-art shifted rapidly toward Transformer-based models [12, 13]. Despite their effectiveness, supervised deep learning models require substantial labeled datasets, which are often unavailable for specific languages or domains. This limitation has led to the exploration of LLMs and prompting techniques. Recent studies by [14] and [15] compared fine-tuned encoders (like XLM-RoBERTa) against generative models (like GPT-3.5) in few-shot scenarios. Their findings suggest a clear trade-off: while fine-tuned models excel when data is abundant, prompt-based approaches offer a more robust solution in low-resource and few-shot settings.

Despite the notable performances reached by state-of-the-art ATE techniques, there is still room for improvement, especially in multilingual and domain-specific contexts [2]. Moreover, performance remains modest on complex datasets, and most approaches still struggle with domain sensitivity.

Evaluation of ATE techniques relies on benchmark corpora such as GENIA [16] and ACL RD-TEC [17], for English, or the multilingual ACTER [18], for English, French, and Dutch. Although Italian has been underrepresented in ATE benchmarks, existing bilingual corpora such as BitterCorpus [19] and MAGMATiC [20] include annotated terms. However, they are primarily designed to evaluate domain-specific machine translation rather than term extraction.

Several shared tasks addressed ATE. Notably, the TermEval 2020 Shared Task on Automatic Term Extraction [21] compared a range of different techniques using the multilingual ACTER corpus. More recently, shared tasks have evolved beyond simple term recognition to emphasize deep semantic understanding and domain-specific relevance. For example, SimpleText Task 2 at CLEF 2024 [22, 23] focused on identifying and explaining complex concepts in scientific abstracts. Participants were asked not only to extract challenging terms but also to generate informative definitions or explanations. Similarly, the ongoing GutBrain Interplay Task3 [24] at BioASQ CLEF 2025 [25] targets the extraction of structured biomedical knowledge related to the gut-brain axis. Its subtasks include term span classification and relation identification, emphasizing fine-grained categorisation and concept linking in highly specialized biomedical texts.

3. Motivation

ATE systems support terminologists, translators, and technical communicators in building and maintaining controlled vocabularies and thesauri for various specialized domains. Moreover, these systems contribute to improving domain-specific language resources used by AI systems for regulatory compliance, automated indexing, and smart information retrieval.

The international interest in ATE is evidenced by the increasing number of European and global research initiatives. These range from conferences and summer schools, such as MDTT² and TSS,³ to

²<https://mdtt2026.dei.unipd.it/>

³https://www.termnet.org/english/products_service/summer_school.php

international ATE shared tasks [21, 23], and the development of numerous gold-standard datasets in English and other languages [16, 17, 18].

Despite the growing momentum in multilingual terminology extraction, the availability of standardized evaluation benchmarks for Italian remains limited. While many shared tasks have been organized for English and select other languages, Italian still lacks a well-defined evaluation framework and publicly available datasets to foster comparative research.

Within this landscape, ATE-IT aims to advance ATE research by introducing a dedicated benchmark for the Italian language, focused on the domain of waste management.

Moreover, by combining term extraction with term variants clustering, ATE-IT aligns with ongoing efforts in terminology standardization, machine translation adaptation, and knowledge base population. Crucially, the proposed evaluation framework facilitates comparability and fosters the application of zero-shot and few-shot learning methods, which are fundamental for low-resource languages like Italian. The task will also promote methodological comparisons between rule-based systems, statistical models, and LLM-based architectures in the context of under-resourced institutional Italian.

4. Definition of the Task

The ATE-IT shared task comprises two subtasks of increasing complexity: Term Extraction and Term Variants Clustering. Both subtasks are designed to be linguistically and computationally challenging. The former requires models to generalize from sparse domain-specific training examples. The latter requires semantic comparison and abstraction over morphologically and syntactically diverse variants.

4.1. Subtask A: Term Extraction

Participants receive a set of sentences drawn from a specialized corpus related to municipal waste management. For each sentence, the goal is to identify and extract the terms that are relevant to the waste management domain. Terms may consist of single words (single-word terms) or multiword expressions (multi-word terms), including nouns, verbs, and adjectives.

4.2. Subtask B: Term Variants Clustering

From the list of unique extracted terms, participants are then required to cluster together those terms that refer to the same underlying concept. For example, “raccolta porta a porta” and “raccolta domiciliare” should be placed in the same cluster. Each cluster should represent a single, coherent concept within the waste management domain. This subtask focuses on synonymy, lexical variation, and compositional semantics.

5. Dataset

The dataset for the Term Extraction subtask comprises sentences paired with the corresponding waste management terms. This dataset is partitioned into a training set of 2,308 sentences, a development set of 577, and a test set of 1,142.

The dataset for the Term Variants Clustering subtask is derived directly from the unique terms identified in the extraction task. Within this dataset, each term is mapped to a cluster ID representing a single, coherent concept in the waste management domain. The clustering data includes 713 terms for training and 242 for development, with a test set of 378. Notably, 118 terms from the test set also appear in either the training or development sets.

Both the training and development sets are sourced from the publicly available ItaIst-TermRifiuti corpus and the ItaIst-WasteLexicon termbase [26, 27]. Conversely, the test set is specifically designed for the ATE-IT task.

Table 1
ATE-IT dataset composition.

Subtask	Split	Size	Items	Geographic and Temporal Coverage
Term Extraction	Training	2,308	Sentences	Campania 2005-2023
	Development	577	Sentences	Campania 2005-2023
	Test	1,142	Sentences	Italy 2025
Term Variants Clustering	Training	713	Terms	Campania 2005-2023
	Development	242	Terms	Campania 2005-2023
	Test	378	Terms	Italy 2025

5.1. Training and Development Sets

The training and development sets are derived from the ItaIst-TermRifiuti corpus and the ItaIst-WasteLexicon termbase [26, 27]. ItaIst-TermRifiuti is a stratified sample of the broader ItaIst-DdAC_GRU corpus [28], which contains institutional texts regarding municipal waste management. The corpus is carefully balanced between:

- **Administrative acts:** Ordinances, service charters, and tenders.
- **Informative texts:** Public notices, guides, and press releases.

This balance facilitates the study of terminological variation across different registers and target audiences. Four trained annotators manually identified terminological units and categorised them into domains such as waste management, law/administration, and environment.

The ItaIst-WasteLexicon termbase provides the conceptual backbone for the dataset, containing about 950 terms organised within a framework that encodes relations such as generic/specific and comprehensive/partitive. Concepts are further enriched with definitions sourced from European and Italian legislation.

In the final Term Extraction dataset, terms are associated with a sentence only if they were identified by at least one annotator and exist within ItaIst-WasteLexicon. Consequently, the dataset may exhibit certain inconsistencies. These are intentionally preserved to provide a “realistic” manually annotated environment, challenging participating systems to demonstrate robustness and generalizability to real-world data.

5.2. Test Set

The test set was specifically curated as a benchmark for the ATE-IT task. To evaluate system generalisation, it incorporates more recent documents (primarily from 2025) and covers a broader geographic scope than the training data. Specifically, while the ItaIst-TermRifiuti dataset is composed primarily of documents from the Campania region, the test set consists of documents collected from 32 municipalities that were sampled with probability proportional to their population, promoting nationwide coverage of Italy.

The initial annotation of the test set was performed by 59 students from the Translation-oriented Terminography of the Department of Linguistic and Literary Studies of the University of Padova. Each student was assigned a unique corpus segment and instructed to adhere to the annotation guidelines already established for the ItaIst-TermRifiuti dataset. Subsequently, sentences were sampled to ensure that the register balance remained consistent with the training and development sets. To finalize the corpus, the authors conducted a validation and correction phase. Through discussion, annotation principles were harmonized to resolve inconsistencies and establish a reliable gold standard.

The test set for the Term Variants Clustering subtask was generated from the list of unique terms identified within this corpus. The terms were then manually clustered by the three authors after a discussion phase.

5.3. Format

The dataset for the term extraction task is a collection of records where each entry maps a specific sentence to its corresponding domain-specific terms. Each record contains the source metadata (consisting of a unique `document_id`, a `paragraph_id`, and a `sentence_id`) alongside the raw `sentence_text`. The target annotations are encapsulated in a `terms` field, which lists the single-word and multi-word terms identified as domain-relevant concepts within that sentence. To maintain technical consistency, the dataset adheres to a “longest match” constraint, where nested terms are excluded in favour of the most complete expression (e.g., “impianto di trattamento rifiuti” is extracted while its constituent “trattamento rifiuti” is omitted). Furthermore, all terms are normalised to lowercase and appear without duplicates for any given sentence.

This schema is implemented in both CSV and JSON formats. Notably, in the CSV format, there is a separate row for each term, whereas in the JSON format, the terms are stored within a `term_list` array.

6. Evaluation Measures

The Term Extraction subtask is evaluated using two separate scores: Micro F1 score [29], which evaluates Precision and Recall across all term occurrences in the dataset, and Type F1 score, which assesses the ability to identify unique term types correctly. The Term Variants Clustering subtask is evaluated by using BCubed F1 [30, 31], a metric that assesses clustering quality.

6.1. Evaluation of Subtask A

To provide a comprehensive evaluation of Subtask A, we distinguished between two separate capabilities of the system: first, its ability to identify individual term mentions as they appear in running text, and second, its ability to successfully extract a unique set of terms from the corpus as a whole. Consequently, we produced two distinct rankings based on Micro F1 and Type F1 scores to reflect these separate goals.

Micro F1 is calculated by aggregating the counts of true positives, false positives, and false negatives across all sentences s in the dataset D . Specifically:

- TP_s : number of terms correctly extracted from sentence s ;
- FP_s : number of terms incorrectly extracted from sentence s ;
- FN_s : number of gold standard terms in s that were missed.

The micro-averaged Precision, Recall, and F1 are defined as shown in equation (1).

$$P_m = \frac{\sum TP_s}{\sum (TP_s + FP_s)}, \quad R_m = \frac{\sum TP_s}{\sum (TP_s + FN_s)}, \quad F1_m = \frac{2 \cdot P_m \cdot R_m}{P_m + R_m} \quad (1)$$

The Type F1 score is computed over the set of unique term types (i.e., distinct term forms appearing at least once in the dataset). We define the following variables:

- TP_t : number of unique extracted terms that match the gold standard;
- FP_t : number of unique extracted terms that do not appear in the gold standard;
- FN_t : number of unique gold standard terms that were not extracted.

The Precision, Recall, and F1 for term types are defined as in equation (2).

$$P_t = \frac{TP_t}{TP_t + FP_t}, \quad R_t = \frac{TP_t}{TP_t + FN_t}, \quad F1_t = \frac{2 \cdot P_t \cdot R_t}{P_t + R_t} \quad (2)$$

You are an automatic term extraction agent. You will receive a list of sentences as input. Your role is to extract waste management terms from the sentences. Output a list of terms for each sentence. strictly adhere to the Example Output Format: Sentence 1: [term1; term2] Sentence 2: [term5; term6] ... Instructions: * Extract only terms, ignore named entities; * Do not extract nested terms; * Extract only terms related to waste management; * If no terms, output an empty list []; * You must output 20 lists of terms.	You are a term clustering agent. You will receive a list of term clusters and a list of unclustered terms related to municipal waste management. Your task is to cluster together exact synonyms. Each cluster must represent a single concept. Output: Return the list of clusters with the newly added terms. Each cluster on a new line. Instructions: * Group terms by meaning, not form. Use their lemma . * Focus on meaning within waste management context. * If a term does not belong to a cluster, create a new cluster.
--	---

Figure 1: System prompts used in the baseline implementation for Subtasks A and B.

6.2. Evaluation of Subtask B

To assess the quality of the clustering in Subtask B, we employed the BCubed metric. This measure is calculated by computing Precision and Recall at the item level and subsequently averaging these scores across all items. Crucially, participants were required to cluster the unique terms extracted by their own systems in Subtask A. Consequently, the set of items in the predicted clustering does not perfectly align with the gold standard. To address this issue, our evaluation framework explicitly accounts for the discrepancy between the two sets.

To compute the BCubed scores, we define the following variables:

- N_{pred} : the total number of elements in the predicted clustering;
- N_{gold} : the total number of elements in the gold clustering;
- $C(x)$: the predicted cluster containing element x (if x is not present in the predicted clustering, $C(x) = \emptyset$);
- $L(x)$: the gold cluster containing element x (if x is not present in the gold clustering, $L(x) = \emptyset$).

For each item x , the item-level Precision and Recall are calculated as in equation (3).

$$P(x) = \frac{|\{y \in C(x) : L(y) = L(x)\}|}{|C(x)|}, \quad R(x) = \frac{|\{y \in L(x) : C(y) = C(x)\}|}{|L(x)|} \quad (3)$$

Finally, the global scores are derived by averaging these item-level values over the respective total counts, and the harmonic mean is taken to produce the final F1 score, as shown in equation (4).

$$P_{b^3} = \frac{1}{N_{pred}} \sum_x P(x), \quad R_{b^3} = \frac{1}{N_{gold}} \sum_x R(x), \quad F1_{b^3} = \frac{2 \cdot P_{b^3} \cdot R_{b^3}}{P_{b^3} + R_{b^3}} \quad (4)$$

7. Baseline

To quantify the task’s complexity and provide a reference point for all participating systems, we provide a baseline system built on the gemini-2.5-flash model in a zero-shot setup. The choice of this model is motivated by its status as a state-of-the-art LLM with robust zero-shot capabilities. It reflects current general-purpose AI performance, allowing for a clear comparison between zero-shot LLM prompting and the specialised, domain-tuned approaches that the task seeks to promote.

Table 2
Results for Subtask A: Term Extraction.

Team	Method	P_m	R_m	$F1_m$	P_t	R_t	$F1_t$	Rank ($F1_m$)	Rank ($F1_t$)
SMTE	BERT+spaCy	.656	.577	.614	.645	.529	.581	1	1
TrietNLP	RoBERTa+CRF	.634	.568	.599	.599	.545	.571	2	2
TEXA (run 1)	BERT	.617	.578	.597	.576	.460	.512	3	5
OA-TE (run 2)	BERT+CRF	.581	.522	.550	.569	.492	.528	4	4
MinseokKIM	CRF	.569	.476	.519	.654	.444	.529	5	3
OA-TE (run 1)	BERT+CRF	.560	.446	.497	.595	.415	.489		
Juliette Tonneau	CRF	.555	.448	.496	.561	.447	.498	6	6
TEXA (run 2)	Gemini (zero-shot)	.471	.514	.492	.425	.489	.455		
Peacemaker	BERT	.497	.476	.486	.430	.455	.442	7	8
TermNinjas	BERT	.489	.395	.437	.528	.404	.458	8	7
Valentinitalie	Random Forest	.364	.473	.411	.707	.262	.382	9	9
<i>baseline</i>	Gemini (zero-shot)	.497	.559	.526	.435	.508	.469		

The Term Extraction baseline model was instructed to identify and extract domain-specific terms from batches of 20 sentences. The system prompt, illustrated in Figure 1, establishes the extraction rules and the required output structure, while the user prompt provides 20 sentences per call.

For Term Variants Clustering, the model was instructed to group synonyms by comparing batches of 20 terms against an existing set of clusters. The system prompt, illustrated in Figure 1, provides the clustering rules while the user prompt feeds the current state of the clusters, followed by the 20 unclustered terms to be processed.

8. Participating Systems

A total of 9 teams participated in the ATE-IT shared task. For Subtask A, 11 runs were submitted, while Subtask B saw a lower participation rate with only 2 submitted runs.

The majority of runs for Subtask A (7 out of 11) leveraged Transformer-based models, specifically the Italian versions of BERT and the multilingual XLM-RoBERTa. Within this group, two teams (TrietNLP [32] and OA-TE [33]) placed a CRF (Conditional Random Field) layer on top of the model to better capture global dependencies in term spans, while the other four (SMTE [34], TEXA [35], Peacemaker [36], and TermNinjas [37]) relied on a token classification layer. The winning system, SMTE, combined BERT with a spaCy-based NER pipeline through a specific merging and vocabulary filtering strategy. In contrast, teams such as MinseokKIM [38] and Juliette Tonneau [39] relied on traditional CRF classifiers, focusing on feature engineering. A hybrid approach was proposed by Valentinitalie [40], which utilised a Random Forest classifier fed by candidates extracted through a combination of rule-based patterns and LLM prompting. Finally, zero-shot prompting was explored by the TEXA team.

Only two teams, TrietNLP and TermNinjas, participated in Subtask B. TrietNLP proposed a pipeline that integrates pre-clustering based on Levenshtein distance with prompting. Similarly, TermNinjas employed lemmatisation during pre-clustering and subsequently merged clusters based on word embedding similarity.

9. Results

The results for Subtask A (Term Extraction) and Subtask B (Term Variants Clustering) are presented in Table 2 and Table 3, respectively.

Table 3

Results for Subtask B: Term Variants Clustering.

Team	Method	P_{b^3}	R_{b^3}	$F1_{b^3}$	Rank ($F1_{b^3}$)
TrietNLP	levenshtein+Gemini	.528	.378	.441	1
TermNinjas	lemmatization+embedding	.390	.333	.359	2
<i>baseline</i>	Gemini (zero-shot)	.177	.396	.245	

9.1. Results for Subtask A

In Subtask A, overall performance was robust, with several systems exceeding an $F1_m$ score of .55. The system submitted by SMTE achieved the highest performance across both micro- and type-based metrics ($F1_m = .614$, $F1_t = .581$). This result highlights the efficacy of integrating BERT-based architectures with specialised NLP pipelines, such as spaCy, for refined terminology extraction.

While TrietNLP and TEXA recorded comparable Micro F1 scores (.599 and .597, respectively), their performance on unique term types ($F1_t$) diverged by a significant margin of .059. This discrepancy suggests that while standard BERT models are effective at identifying frequent terminological mentions, the RoBERTa+CRF architecture employed by TrietNLP is better suited for identifying the “long tail” of rare variants and unseen terms.

The performance of Valentinitalie is noteworthy; despite recording the lowest overall F1 scores ($F1_m = .411$, $F1_t = .382$), it achieved the highest Type Precision across all participants ($P_t = .707$). However, subsequent analysis reveals that the extracted terms were almost exclusively present in the training set. This indicates a conservative extraction strategy that prioritized precision over the ability to generalise to novel terminology in the test set.

General trends across the submissions indicate that supervised approaches typically achieved higher Precision than Recall. Conversely, zero-shot models favoured Recall, which may suggest a superior ability to identify term boundaries and adhere to the strict “longest match” constraint. Finally, the consistent disparity between Micro F1 and Type F1 scores across all teams confirms that identifying frequent term mentions remains significantly easier than discovering the full diversity of unique terms within a specialised corpus.

9.2. Results for Subtask B

Participation in the Term Variants Clustering subtask was significantly lower than in Subtask A, with only two teams submitting valid runs, reflecting the higher complexity of the challenge.

The best performance was achieved by TrietNLP, which recorded a BCubed F1 score of .441. Their hybrid approach, combining Levenshtein distance for morphological pre-clustering with a Gemini-based component for semantic aggregation, proved effective in maintaining high purity within clusters, as evidenced by the highest Precision score ($P_{b^3} = .528$).

TermNinjas ranked second with an F1 score of .359. Their strategy, which relied on lemmatization and word embedding similarity, struggled to match the Precision of the top system, achieving a P_{b^3} of .390.

Notably, the zero-shot baseline achieved the highest Recall ($R_{b^3} = .396$) but the lowest Precision ($P_{b^3} = .177$). This inverse relationship suggests that the baseline tended to over-cluster, aggressively grouping terms together. This behavior underscores the primary challenge of the task: effectively distinguishing exact synonyms from other semantic relations, thereby ensuring that hyponyms and hypernyms are not erroneously merged into the same concept cluster.

10. Discussion

The results of the ATE-IT shared task provide a snapshot of the current capabilities in Italian ATE. Beyond the individual rankings, several major trends emerged: the reliability of deep learning, the necessity of structural decoding for accurate term boundary detection, the challenge of generalisation over diatopic and diachronic variations, and the emergence of hybrid approaches for term variants clustering.

10.1. Deep learning vs. LLMs prompting

The campaign confirms the resilience of standard supervised deep learning architectures in the era of LLMs, aligning with findings already established in the broader ATE literature [5]. While the naive zero-shot baseline achieved a respectable performance, it was consistently outperformed by more lightweight systems based on BERT and RoBERTa architectures.

This result reinforces that for ATE, where a training set exists, fine-tuned models offer a better alternative to prompting LLMs, making them suited for real-world applications, where speed and cost are key factors.

10.2. Term boundary detection

Moreover, results suggest that Transformer architectures alone are often insufficient for accurate term boundary detection. In fact, the top-performing systems augmented it with structural decoding mechanisms like the spaCy-based pipeline of SMTE, or the CRF layer integrated by TrietNLP. These choices proved decisive in enforcing the “longest match” constraint.

Terminology in the waste management sector is highly compositional (e.g., “raccolta” vs. “raccolta differenziata” vs. “raccolta differenziata porta a porta”). Therefore, purely statistical models may fragment these multi-word terms, whereas systems with explicit boundary modelling or post-processing heuristics successfully captured the full syntactic structure of these terminological units.

10.3. Diatopic and diachronic variations

Another crucial insight from this evaluation is the significant impact of temporal and geographic shifts on model performance. During the development phase, several systems reportedly achieved F1 scores exceeding .70; however, performance dropped noticeably on the test set ($F1_m \approx .60$ for the best system). This decline highlights the difficulty of the realistic scenario proposed by ATE-IT.

This suggests that even robust deep learning models tend to overfit to specific regional denominations or temporal periods. The challenge for future Italian ATE lies in improving its generalization capabilities, since terminology may exhibit significant diatopic variation and evolve diachronically.

10.4. The “long tail” challenge

The consistent disparity between Micro F1 and Type F1 scores across all teams indicates that identifying frequent term mentions is significantly easier than discovering the full diversity of unique terms.

This “long tail” problem is particularly relevant for the intended application of ATE systems. For a terminologist or a translator, high-frequency terms (e.g., “rifiuto”, “organico”) are often already known and documented. The real value of an extraction system lies in its ability to surface rare, highly specific, or emerging terms (e.g., “presidi sanitari monouso”, “pseudo-edili”).

10.5. Hybrid approaches for semantic clustering

Finally, the results of Subtask B shed light on the complexity of Term Variants Clustering. The baseline performance was low ($F1_{b3} = .245$), confirming that simple zero-shot prompting struggles to effectively group domain-specific synonyms without guidance. As observed in the results, the main challenge lies

in the strict definition of synonymy required by the task; systems must navigate the subtle semantic boundary that separates synonyms from hypernyms and hyponyms, a distinction where the baseline frequently faltered.

The winning team, TrietNLP, achieved a significantly higher score by employing a hybrid approach. The result suggests that an effective strategy for addressing this task appears to be a composite pipeline: using symbolic methods to handle surface-level morphological variations and leveraging the reasoning capabilities of LLMs to enforce semantic equivalence.

11. Conclusion

The ATE-IT shared task addressed the need for a dedicated benchmark for Automatic Term Extraction in Italian. By introducing a dataset characterized by marked terminological variation, the campaign allowed for a comparative analysis of diverse computational approaches.

The results from the participation of 9 teams and the evaluation of 13 runs point to three main findings. First, regarding Term Extraction, fine-tuned Transformer architectures outperformed the naive zero-shot baseline provided for the task. This indicates that while simple prompting strategies are insufficient for high-precision extraction, supervised models remain a reliable standard for this specific workload. However, given the rapid evolution of generative AI, these results should not discount the potential of LLMs; rather, they suggest that future research should investigate more advanced prompting techniques to bridge the gap with dedicated supervised systems.

Second, the performance gap observed between the development and test sets indicates that geographic and temporal generalisation remains a complex issue. Current models show a tendency to overfit to the specific term variants of the training data, resulting in lower F1 scores when applied to documents from different municipalities.

Finally, the low baseline results in Term Variants Clustering suggest that synonym clustering requires more than simple semantic similarity. The most effective system adopted a hybrid strategy, combining symbolic methods for morphological matching with LLMs for semantic disambiguation.

We hope the open release of the ATE-IT dataset will support the community in developing more robust Automatic Term Extraction methodologies specifically tailored to the complexities of the Italian language.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini in order to: Drafting content, Improve writing style, and Abstract generation. Further, the authors used Gemini for equations 1 to 4 in order to: Formatting assistance. Finally, the authors used Gemini and Grammarly in order to: Grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] K. Kageura, B. Umino, Methods of automatic term recognition: A review, *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.
- [2] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. doi:10.1093/llc/fqad030.
- [3] B. Jehangir, S. Radhakrishnan, R. Agarwal, A survey on Named Entity Recognition — datasets, tools, and methodologies, *Natural Language Processing Journal* 3 (2023) 100017. doi:10.1016/j.nlp.2023.100017.

- [4] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [5] J. C. Blandón Andrade, C. M. Medina Otálvaro, C. M. Zapata Jaramillo, A. Morales Ríos, Approaches, Tools, Algorithms, and Methods for Automatic Term Extraction: A Systematic Literature Mapping, *Journal of Intelligent & Fuzzy Systems* (2025) 18758967251392652. doi:10.1177/18758967251392652.
- [6] N. A. Astrakhantsev, D. G. Fedorenko, D. Yu. Turdakov, Methods for automatic term recognition in domain-specific text collections: A survey, *Programming and Computer Software* 41 (2015) 336–349. doi:10.1134/S036176881506002X.
- [7] J. S. Justeson, S. M. Katz, Technical terminology: Some linguistic properties and an algorithm for identification in text, *Natural Language Engineering* 1 (1995) 9–27. doi:10.1017/S1351324900000048.
- [8] A. L. Meyers, Y. He, Z. Glass, J. Ortega, S. Liao, A. Grieve-Smith, R. Grishman, O. Babko-Malaya, The Termolator: Terminology Recognition Based on Chunking, Statistical and Search-Based Scores, *Frontiers in Research Metrics and Analytics* 3 (2018). doi:10.3389/frma.2018.00019.
- [9] K. Ahmad, L. Gillam, L. Tostevin, University of surrey participation in TREC8: weirdness indexing for logical document extrapolation and retrieval (WILDER), in: E. M. Voorhees, D. K. Harman (Eds.), Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999, volume 500-246 of *NIST Special Publication*, National Institute of Standards and Technology (NIST), 1999. URL: <http://trec.nist.gov/pubs/trec8/papers/surrey2.pdf>.
- [10] P. Drouin, Term extraction using non-technical corpora as a point of leverage, *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 9 (2003) 99–115. doi:10.1075/term.9.1.06dro.
- [11] M. Kucza, J. Niehues, T. Zenkel, A. Waibel, S. Stüker, Term Extraction via Neural Sequence Labeling a Comparative Evaluation of Strategies Using Recurrent Neural Networks, in: Proc. Interspeech 2018, 2018, pp. 2072–2076. doi:10.21437/Interspeech.2018-2017.
- [12] A. Hazem, M. Bouhandi, F. Boudin, B. Daille, TermEval 2020: TALN-LS2N System for Automatic Term Extraction, in: B. Daille, K. Kageura, A. R. Terryn (Eds.), Proceedings of the 6th International Workshop on Computational Terminology, European Language Resources Association, Marseille, France, 2020, pp. 95–100.
- [13] C. Lang, L. Wachowiak, B. Heinisch, D. Gromann, Transforming Term Extraction: Transformer-Based Approaches to Multilingual Term Extraction Across Domains, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online, 2021, pp. 3607–3620. doi:10.18653/v1/2021.findings-acl.316.
- [14] S. Banerjee, B. R. Chakravarthi, J. P. McCrae, Large Language Models for Few-Shot Automatic Term Extraction, in: A. Rapp, L. Di Caro, F. Mezziane, V. Sugumaran (Eds.), *Natural Language Processing and Information Systems*, Springer Nature Switzerland, Cham, 2024, pp. 137–150. doi:10.1007/978-3-031-70239-6_10.
- [15] H. T. H. Tran, C.-E. González-Gallardo, J. Delaunay, A. Doucet, S. Pollak, Is Prompting What Term Extraction Needs?, in: E. Nöth, A. Horák, P. Sojka (Eds.), *Text, Speech, and Dialogue*, Springer Nature Switzerland, Cham, 2024, pp. 17–29. doi:10.1007/978-3-031-70563-2_2.
- [16] J. D. Kim, T. Ohta, Y. Tateisi, J. Tsujii, Genia corpus - a semantically annotated corpus for bio-textmining, in: *Bioinformatics*, volume 19, Oxford University Press, 2003. doi:10.1093/bioinformatics/btg1023.
- [17] B. Qasemizadeh, A.-K. Schumann, The acl rd-tec 2.0: A language resource for evaluating term extraction and entity recognition methods, in: N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), European Language Resources Association (ELRA), 2016, pp. 1862–1868.

- [18] A. Rigouts Terryn, V. Hoste, E. Lefever, In no uncertain terms: A dataset for monolingual and multilingual automatic term extraction from comparable corpora, *Language Resources and Evaluation* 54 (2020) 385–418. doi:10.1007/s10579-019-09453-9.
- [19] M. Arcan, M. Turchi, S. Topelli, P. Buitelaar, Enhancing statistical machine translation with bilingual terminology in a cat environment, in: Y. Al-Onaizan, M. Simard (Eds.), *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track*, Association for Machine Translation in the Americas, 2014, pp. 54–68.
- [20] R. Scansani, L. Bentivogli, S. Bernardini, A. Ferraresi, Magmatic: A multi-domain academic gold standard with manual annotation of terminology for machine translation evaluation, in: M. Forcada, A. Way, B. Haddow, R. Sennrich (Eds.), *Proceedings of Machine Translation Summit XVII: Research Track*, European Association for Machine Translation, 2019, pp. 78–86.
- [21] A. Rigouts Terryn, V. Hoste, P. Drouin, E. Lefever, TermEval 2020: Shared Task on Automatic Term Extraction Using the Annotated Corpora for Term Extraction Research (ACTER) Dataset, in: B. Daille, K. Kageura, A. R. Terryn (Eds.), *Proceedings of the 6th International Workshop on Computational Terminology*, European Language Resources Association, Marseille, France, 2020, pp. 85–94.
- [22] L. Ermakova, E. SanJuan, S. Huet, H. Azarbondy, G. M. Di Nunzio, F. Vezzani, J. D’Souza, J. Kamps, Overview of the CLEF 2024 SimpleText Track, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, G. M. Di Nunzio, L. Soulier, P. Galuščáková, A. García Seco de Herrera, G. Faggioli, N. Ferro (Eds.), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Springer Nature Switzerland, Cham, 2024, pp. 283–307. doi:10.1007/978-3-031-71908-0_13.
- [23] G. M. Di Nunzio, F. Vezzani, V. Bonato, H. Azarbondy, J. Kamps, L. Ermakova, Overview of the CLEF 2024 SimpleText Task 2: Identify and Explain Difficult Concepts, in: G. Faggioli, N. Ferro, P. Galuščáková, A. G. S. de Herrera (Eds.), *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2024)*, volume 3740 of *CEUR Workshop Proceedings*, CEUR, Grenoble, France, 2024, pp. 3129–3146.
- [24] M. Martinelli, G. Silvello, V. Bonato, G. M. Di Nunzio, N. Ferro, O. Irrera, S. Marchesin, L. Menotti, F. Vezzani, Overview of GutBrainIE@CLEF 2025: Gut-Brain Interplay Information Extraction, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025)*, volume 4038 of *CEUR Workshop Proceedings*, CEUR, Madrid, Spain, 2025, pp. 65–98.
- [25] A. Nentidis, G. Katsimpras, A. Krithara, M. Krallinger, M. R. Ortega, N. Loukachevitch, A. Sakhovskiy, E. Tutubalina, G. Tsoumakas, G. Giannakoulas, A. Bekiaridou, A. Samaras, G. M. Di Nunzio, N. Ferro, S. Marchesin, L. Menotti, G. Silvello, G. Paliouras, BioASQ at CLEF2025: The Thirteenth Edition of the Large-Scale Biomedical Semantic Indexing and Question Answering Challenge, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), *Advances in Information Retrieval*, Springer Nature Switzerland, Cham, 2025, pp. 407–415. doi:10.1007/978-3-031-88720-8_61.
- [26] N. Cirillo, D. Vellutino, D. Nicoletti, E. Sabarese, B. Rubino, Itaist_gru, 2025. URL: <https://doi.org/10.5281/zenodo.15173712>. doi:10.5281/zenodo.15173712.
- [27] N. Cirillo, *Risorse Linguistiche Digitali per la Transizione Verde*, Phd thesis, University of Salerno, 2025.
- [28] D. Vellutino, N. Cirillo, Corpus «itaist»: Note per lo sviluppo di una risorsa linguistica per lo studio dell’italiano istituzionale per il diritto di accesso civico, *Italiano LinguaDue* 16 (2024) 238–250.
- [29] R. Verborgh, M. Röder, R. Usbeck, A.-C. Ngonga Ngomo, Gerbil – benchmarking named entity recognition and linking consistently, *Semant. Web* 9 (2018) 605–625. URL: <https://doi.org/10.3233/SW-170286>. doi:10.3233/SW-170286.
- [30] A. Bagga, B. Baldwin, Entity-based cross-document coreferencing using the vector space model, in: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998. URL: <https://aclanthology.org/C98-1012/>.
- [31] E. Amigó, J. Gonzalo, J. Artiles, F. Verdejo, A comparison of extrinsic clustering evaluation metrics based on formal constraints, *Information Retrieval* 12 (2009) 461–486. doi:10.1007/

- [32] N. M. Triet, D. Van Thin, Trietnlp at ate-it: A hybrid pipeline for italian waste management terminology analysis, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [33] O. Arab, G. M. Di Nunzio, Oate at ate-it: A hybrid approach for automatic terminology extraction in italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [34] S. Maule, G. M. Di Nunzio, Smte at ate-it: Ensemble term extraction with italian bert, spacy, and vocabulary-based filtering, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [35] D. Smirnov, B. Khashechian, G. M. Di Nunzio, Dsbkte at ate-it: From token classification to zero-shot generation: Two approaches to italian ate at evalita 2026, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [36] M. Bakhtiyarzadeh, H. Bayrami Asl Tekanlou, J. Razmara1, Peacemaker at ate-it: Automatic term extraction from italian text for waste management data using encoder model, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [37] Z. Hasnain, F. Ahmed Siddque, Termninjas at ate-it: Overview of the term extraction and term variants clustering task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [38] M. Kim, G. M. Di Nunzio, Mkte at ate-it: Crf-based term extraction for italian waste management documents, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [39] J. Tonneau, G. M. Di Nunzio, Jtte at ate-it: A crf model with contextual embeddings, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [40] V. G. L. Vandervoort, G. M. Di Nunzio, Vvte at ate-it: From candidates to terms: Hybrid italian ate with dependency heuristics, gemini, and random forest filtering, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

DSBKTE at ATE-IT: From Token Classification to Zero-Shot Generation: Two Approaches to Italian ATE at EVALITA 2026

Danil Smirnov^{1,*}, Bitu Khashechian^{1,*} and Giorgio Maria Di Nunzio¹

¹Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

This paper presents our participation in Subtask A of the ATE-IT shared task at EVALITA 2026, which targets sentence-level Automatic Terminology Extraction from Italian administrative documents in the waste-management domain. The dataset includes a balanced collection of administrative acts (e.g., municipal regulations, service charters, tenders) and informative texts (e.g., public notices), combining formal institutional language with less rigid informational prose. These heterogeneous documents are characterized by syntactically complex sentences and dense specialized terminology. Given an input sentence, systems must identify all domain-relevant single- and multiword terms and output them in a predefined JSON format, requiring generalization beyond a fixed vocabulary. We investigate and compare two strategies: (i) a supervised neural approach based on an Italian BERT model fine-tuned for token-level term span detection, and (ii) a zero-shot approach using large language models to extract terminology without task-specific training. We report official results and provide an analysis of typical errors to highlight the relative strengths and limitations of supervised versus zero-shot methods for terminology extraction in complex institutional texts.

Keywords

Automatic Terminology Extraction, Information Extraction, BERT, Large Language Models, Italian Administrative Texts

1. Introduction

Automatic Terminology Extraction (ATE) is the task of automatically identifying domain-specific terms from textual data [1]. These terms typically correspond to multiword expressions or noun phrases that denote relevant technical, legal, or institutional concepts within a specific domain. Accurate terminology extraction plays a crucial role in many downstream Natural Language Processing applications, including information retrieval, document classification, ontology construction, and knowledge base population.

The ATE-IT [2] shared task, organized within the EVALITA 2026 evaluation campaign [3], focuses on terminology extraction from Italian administrative documents in the waste-management domain. The dataset is composed of heterogeneous official texts such as municipal regulations, service charters, public tenders, and informational notices issued by local authorities. These documents include both administrative acts and informative texts. Administrative acts typically exhibit formal institutional language and recurring syntactic patterns, whereas informative texts may adopt less rigid stylistic structures. Both document types are characterized by long, syntactically complex sentences and a high density of specialized terminology, making the extraction task particularly challenging.

ATE-IT is structured into two subtasks. Subtask A addresses sentence-level Automatic Term Extraction, where systems are required to identify all relevant domain terms occurring in a given sentence. For each input sentence, the system must output a list of extracted terms following a predefined JSON format. The extracted terms may consist of single words or multiword expressions and are not restricted to a fixed vocabulary, requiring models to generalize beyond surface-level patterns.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ danil.smirnov@studenti.unipd.it (D. Smirnov); bitu.khashechian@studenti.unipd.it (B. Khashechian);
giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

ORCID 0000-0001-7116-9338 (G. M. Di Nunzio)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

In this work, we focus exclusively on Subtask A. We investigate two approaches to term extraction: a supervised neural model based on BERT [4] fine-tuned for token classification, and a zero-shot approach based on large language models. The supervised approach leverages annotated training data to learn precise term boundaries, while the zero-shot method explores the ability of generative models to identify domain terminology without task-specific training. Through this comparison, we aim to analyze the strengths and limitations of supervised and zero-shot strategies for terminology extraction in complex administrative texts.

The remainder of this paper is organized as follows. Section 2 reviews related work on supervised and zero-shot terminology extraction. Section 3 describes the task and dataset. Section 4 presents the system architectures and experimental setup. Section 5 reports quantitative results. Section 6 discusses error patterns, limitations, and future directions. Section 7 concludes the paper.

2. Related Work

Automatic Terminology Extraction (ATE) has been addressed using statistical, linguistic, and supervised learning approaches [5]. Early methods relied on corpus-based termhood measures such as frequency statistics, TF-IDF variants, and heuristics like C-value to handle multiword terms, but they depend on reliable candidate generation and are sensitive to domain-specific phrasing [6].

Linguistic pipelines typically extract noun phrases using POS tagging or parsing. While grammatical constraints can improve precision, their effectiveness strongly depends on the quality of NLP tools and often degrades in administrative texts with complex formatting and enumerations.

More recent work formulates ATE as a supervised sequence labeling task, closely related to Named Entity Recognition. Neural models based on BiLSTM-CRF and, in particular, Transformer architectures such as BERT achieve accurate boundary detection when fine-tuned with BIO labels, though they require careful subword alignment and post-processing [4].

Large language models have recently enabled zero-shot and few-shot ATE via prompt-based inference. These approaches generalize without task-specific training but often exhibit limited controllability, sensitivity to prompt formulation, and tendencies toward over-generation or boundary inconsistency in structured extraction tasks [7, 8]. Models such as Gemini [9] and LLaMA [10] show strong instruction-following capabilities, yet may still generate annotation-inconsistent spans without additional constraints.

In shared-task evaluations such as EVALITA, systems are compared under standardized formats and strict term-level evaluation. Within this setting, supervised token classification and zero-shot LLM-based extraction represent complementary strategies, trading boundary precision for generalization capability [11].

3. Task and Dataset Description

The ATE-IT shared task [2] evaluates Automatic Terminology Extraction systems on Italian administrative documents in the waste-management domain. The dataset is derived from the ItaIst_GRU corpus [12] and consists of a balanced collection of administrative acts and informative texts issued by local authorities. Administrative acts typically exhibit formal institutional language and recurring syntactic patterns, while informative texts may adopt less rigid stylistic structures. Both document types are characterized by a high density of multiword terminology, making boundary detection particularly challenging.

ATE-IT is organized into two subtasks. In this work, we focus on Subtask A, which addresses sentence-level term extraction. Given an input sentence, systems must identify all domain-specific terms appearing in the sentence and output them as a list of strings. Terms may be single words or multiword expressions and are not restricted to a predefined vocabulary.

The dataset is provided in CSV format and split into training, development, and test sets. Each row corresponds to a sentence-term pair; sentences with multiple annotated terms therefore appear in

multiple rows and must be aggregated at the sentence level during preprocessing.

Annotated terms cover a range of waste-management and administrative concepts (e.g., *servizio di raccolta dei rifiuti urbani*, *utenze domestiche*), many of which are long multiword expressions, making boundary detection a key challenge.

Evaluation follows the official ATE-IT protocol, which computes micro-averaged Precision, Recall, and F_1 score at the term level using exact string matching after normalization [11].

4. Description of the system

4.1. System Overview

We compare two approaches to Automatic Terminology Extraction for ATE-IT Subtask A: a supervised sequence labeling model and a zero-shot prompt-based extraction pipeline. The two methods are implemented as independent pipelines to enable a clear and controlled comparison of their behavior and error patterns.

Both systems operate at the sentence level and share the same input/output protocol, producing a list of extracted terms in the JSON format required by the task. The supervised approach is based on a BERT token classification model trained on annotated data and optimized for precise span detection, while the zero-shot approach relies on a large language model and performs term extraction via prompt-based inference without task-specific training. No ranking or confidence estimation is applied, ensuring full compatibility with the official evaluation procedure.

4.2. Supervised BERT-based Term Extraction

The supervised approach formulates Automatic Terminology Extraction as a sequence labeling problem, where each token in a sentence is assigned a label indicating whether it belongs to a domain-specific term. We adopt the BIO tagging scheme, in which labels B-TERM and I-TERM denote the beginning and continuation of a term span, respectively, while O marks tokens outside any term. This formulation enables explicit modeling of term boundaries and allows the extraction of multiword expressions of arbitrary length.

Preprocessing and label construction. During preprocessing, the training data provided in CSV format is converted into sentence-level instances. Since the dataset represents sentences with multiple annotated terms across multiple rows, all annotations sharing the same document, paragraph, and sentence identifiers are aggregated. Each annotated term is then aligned with its corresponding span in the sentence text. Tokenization is performed at the word level, and BIO labels are assigned by matching normalized word sequences against the gold-standard term spans. Tokens that do not belong to any annotated term are labeled as O.

Model architecture. We employ a Transformer-based encoder using a pretrained Italian BERT model, fine-tuned for token classification. The model is implemented using the HuggingFace `AutoModelForTokenClassification` framework [13], which adds a linear classification layer on top of the contextualized token representations produced by BERT [4]. The classifier predicts a probability distribution over the BIO label set for each token in the input sequence.

Subword alignment. As BERT operates on subword units, careful alignment between word-level BIO labels and subword tokens is required. During training, word-level tokens are expanded into subword sequences using the model tokenizer. The BIO label assigned to each word is propagated to its corresponding subwords following standard practices. Special tokens and padding positions are masked during loss computation. At inference time, predicted subword labels are mapped back to word-level labels using the tokenizer alignment information, ensuring consistent reconstruction of term spans.

Training setup. The model is fine-tuned on the ATE-IT Subtask A training split using cross-entropy loss. Training is performed with the HuggingFace Trainer API, using the Adam optimizer and a fixed number of epochs. Hyperparameters such as batch size, learning rate, and maximum sequence length are selected to balance training stability and computational efficiency. No additional domain-adaptive pretraining is applied, allowing us to assess the effectiveness of the base Italian BERT model on administrative texts.

Inference and span reconstruction. During inference, the model predicts a BIO label for each subword token. These predictions are aggregated at the word level, and contiguous B-TERM/I-TERM sequences are merged to form complete term spans. The extracted terms preserve their original surface form as they appear in the input sentence. This reconstruction step is critical for producing outputs that are compatible with the exact string matching used in the official evaluation script.

4.3. Zero-shot LLM-based Term Extraction

We also test a zero-shot terminology extraction pipeline based on large language models (LLMs). In this setting, no task-specific training data is used: the model is prompted to return the domain terms appearing in each input sentence.

Model. Our main system uses Gemini-2.5-Flash [9]. We additionally tested LLaMA 3.1 [10] and GPT-5.1, but Gemini produced the most stable formatting and the best development-set performance among the evaluated LLMs.

Prompting and output format. Sentences were processed in batches, and the model was instructed to extract waste-management and administrative terms and to output one list of terms per sentence (without explanations), matching the Subtask A requirements. We report below the exact prompt template used for all zero-shot experiments. The same instruction was applied uniformly across all batches, without in-context examples.

```
You are an automatic term extraction agent.
You will receive a list of Italian sentences about municipal waste management.
Your role is to extract waste management terms from the sentences.
Output a list of terms for each sentence.
```

Strictly adhere to the Example Output Format:

```
Sentence 1: [term1; term2; term3]
Sentence 2: [term4]
Sentence 3: []
```

Instructions:

- * Extract only domain-relevant terms related to waste management.
- * Terms can be single- or multi-word.
- * Use only lowercase in the output.
- * Do not output nested terms (if you output a longer term, do not also output its inner substring as a separate term).
- * Do not output named entities such as city names or organisation names.
- * If a sentence contains no terms, output an empty list [].
- * Output exactly N lists of terms, one for each input sentence.

Post-processing. We apply lightweight normalization (whitespace and punctuation cleanup) and remove duplicates. No ranking or confidence scoring is used.

4.4. Experimental Setup

All experiments are conducted using the official ATE-IT dataset for Subtask A, which is split into training, development, and test sets. The supervised BERT-based model is trained on the training split and evaluated on the development split. The development data is used for model selection, qualitative analysis, and comparison with the zero-shot approach, while the test split is reserved exclusively for final submission.

The zero-shot LLM-based approach does not rely on annotated training data and is applied directly to the development and test splits using the same extraction pipeline, enabling a direct comparison between supervised and zero-shot methods under identical conditions.

Evaluation metrics. Performance is measured using the official ATE-IT evaluation script for Subtask A, which computes micro-averaged Precision, Recall, and F_1 score at the term level. A prediction is considered correct only if it exactly matches a gold-standard term after normalization; partial matches and overlapping spans are counted as errors, emphasizing strict boundary accuracy.

Implementation details. The supervised model is implemented with the HuggingFace Transformers library and fine-tuned using the Trainer API with cross-entropy loss. Input sequences are truncated or padded to a fixed maximum length, and padding tokens are excluded from loss computation. All experiments are run with a fixed random seed to limit variability.

The zero-shot pipeline is implemented as a prompt-based interface to large language models. Each sentence is processed with a single prompt, and outputs are normalized through post-processing. No in-context examples are provided, ensuring a purely zero-shot setting.

Reproducibility. Preprocessing and model configurations are kept fixed across runs, and evaluation is consistently performed with the official script and normalization settings. While exact reproducibility of zero-shot outputs cannot be guaranteed due to model non-determinism, prompt design and post-processing remain unchanged throughout all experiments.

5. Results

We report results for both the development and test sets using the official ATE-IT evaluation script for Subtask A. The script computes term-level micro-averaged Precision, Recall, and F_1 score by comparing system predictions against gold-standard annotations under a strict exact-match criterion. All predictions are generated in the required JSON format using the pipelines described in the previous sections.

Development set. On the development set, the supervised BERT-based model achieves strong performance, reaching a micro-averaged Precision of 0.73, Recall of 0.67, and an F_1 score of 0.70 (Table 1). These results indicate that the model effectively learns term boundaries from annotated data and maintains a good balance between precision and recall across heterogeneous administrative sentences.

The zero-shot LLM-based approach, evaluated under the same conditions, shows a different performance profile. It attains a Precision of 0.48 and Recall of 0.60, resulting in an F_1 score of 0.53. While recall is relatively high, the lower precision reflects a tendency to over-generate candidate terms or produce imprecise spans, which are penalized by the exact-match evaluation. All zero-shot results on the development set are obtained using the Gemini 2.5-flash model [9].

Test set. Following the release of the official test set, we submitted both systems using the same configurations adopted for development experiments. Table 2 reports the official test-set results, together

Approach	Precision	Recall	F1-score
BERT (supervised)	0.73	0.67	0.70
Zero-shot LLM (Gemini)	0.48	0.60	0.53

Table 1

Term-level micro-averaged Precision, Recall, and F_1 score on the ATE-IT Subtask A development set.

Approach	Micro P	Micro R	Micro F1	Type P	Type R	Type F1
Baseline	0.497	0.559	0.526	0.435	0.508	0.469
BERT (supervised)	0.617	0.578	0.597	0.576	0.460	0.512
Zero-shot LLM (Gemini)	0.471	0.514	0.492	0.425	0.489	0.455

Table 2

Official test-set results for ATE-IT Subtask A. Micro-level metrics evaluate term occurrences, while type-level metrics evaluate distinct term types.

with the baseline provided by the ATE-IT organizers. In addition to term-level (micro) metrics, we also report type-level Precision, Recall, and F_1 , which evaluate coverage of distinct terminology types.

On the test set, the supervised BERT-based model consistently outperforms the baseline across all metrics, improving Micro F_1 from 0.526 to 0.597. This gain is primarily driven by higher precision, indicating more accurate boundary detection and fewer spurious extractions. Type-level results show a similar trend, with the BERT model achieving higher Type F_1 than the baseline, suggesting better coverage of distinct domain terms.

The zero-shot LLM-based approach exhibits stable recall but lower precision, leading to Micro and Type F_1 scores below the baseline. This behavior mirrors the development-set results and confirms that, under strict exact-match evaluation, zero-shot extraction tends to favor coverage over boundary accuracy.

Overall, results on both development and test sets consistently show that supervised sequence labeling with BERT provides the most reliable performance when annotated data is available. Zero-shot LLM-based extraction remains competitive in terms of recall and demonstrates robustness without task-specific training, but its lower precision limits performance under the evaluation criteria adopted in ATE-IT.

6. Discussion

The experimental results confirm that the supervised BERT-based approach provides the most reliable boundary detection under the strict exact-match evaluation adopted in ATE-IT. Leveraging annotated training data, the model achieves strong precision and stable performance across heterogeneous administrative sentence structures, particularly for well-formed multiword terms frequently observed in the corpus. However, its reliance on supervision constrains generalization: rare expressions, highly variable syntactic constructions, and implicitly expressed terminology remain challenging, reflecting the inherent limitations of purely supervised learning in specialized domains.

6.1. Error Analysis

We perform a qualitative error analysis on the development set to identify recurring failure patterns of the supervised BERT-based model and the zero-shot LLM-based extractor. Manual inspection reveals several systematic error categories.

Boundary errors. Both approaches suffer from boundary inaccuracies. The supervised model often truncates long multiword terms with coordination or prepositional attachments, extracting only the core noun phrase. Conversely, the zero-shot approach tends to generate overly long spans by merging

adjacent expressions or including punctuation, especially in enumerations. Although semantically plausible, such errors are penalized by exact-match evaluation.

Over-generation. The zero-shot system frequently over-generates candidate terms in long administrative sentences, including generic or discourse-level nouns not annotated as domain terms. This behavior improves recall but significantly reduces precision, reflecting a preference for semantic salience over annotation consistency.

Missed terms. Both systems occasionally miss relevant single-word or implicit terms, particularly in complex syntactic contexts. For the supervised model, these errors are more common for rare expressions or terms underrepresented in the training data, indicating sensitivity to data sparsity.

Formatting issues. Casing and punctuation inconsistencies affect both approaches and often result in incorrect matches under the strict evaluation protocol, despite limited semantic impact.

Conservative vs. exhaustive behavior. The supervised model may produce empty outputs for difficult sentences, while the zero-shot model almost always returns at least one candidate term, confirming its more exhaustive but less precise behavior.

Overall, the supervised approach provides more consistent and precise boundaries, whereas the zero-shot method offers broader coverage at the cost of increased noise. These complementary error patterns suggest that hybrid strategies could further improve robustness.

6.2. Future Work

The experimental results highlight different performance profiles of supervised and zero-shot approaches for Automatic Terminology Extraction in Italian administrative texts. The supervised BERT-based model achieves higher precision and overall F_1 score, indicating that sequence labeling with annotated data remains the most reliable strategy when accurate term boundaries are required. Its strength lies in consistent extraction of well-formed multiword terms that follow recurring syntactic patterns.

The zero-shot LLM-based approach, while less precise, exhibits higher recall and robustness to previously unseen expressions. This suggests that large language models encode useful domain and linguistic knowledge that can be leveraged without task-specific training. Such behavior is particularly relevant in low-resource scenarios or domains where annotated data is limited or rapidly outdated.

Error analysis shows that many failures are driven by structural properties of the data. Long sentences, complex nominal constructions, and enumerations challenge both approaches in different ways. The supervised model tends to be conservative and may miss relevant terms, whereas the zero-shot model frequently over-generates candidates or produces imprecise boundaries. These complementary error profiles indicate that no single paradigm fully addresses the complexity of the task.

These observations motivate the exploration of hybrid strategies that combine supervised and zero-shot extraction. For example, zero-shot predictions could be used as a fallback when the supervised model produces empty outputs, or as a candidate generation step followed by supervised or rule-based filtering.

Future work may also consider domain-adaptive pretraining of BERT on administrative or legal corpora, span-based extraction models to reduce boundary fragmentation, and the integration of syntactic information to better constrain term spans. For LLM-based methods, further research is needed on improving controllability and boundary consistency through prompt optimization or lightweight adaptation. Overall, combining supervised learning with large language models appears to be a promising direction for improving terminology extraction in complex administrative domains.

6.3. Limitations and Ethical Considerations

The proposed approaches present several limitations. The supervised BERT-based model depends on the availability and quality of annotated data. Although the ATE-IT dataset provides reliable annotations, its limited size and domain scope may restrict generalization to administrative documents with different structure, terminology, or legal context. Rare terms and implicitly expressed domain concepts are particularly prone to being missed.

A further limitation concerns boundary sensitivity. Subtask A evaluation is based on exact string matching, which penalizes even minor boundary deviations such as trailing punctuation, casing mismatches, or partial span truncation. While this criterion enforces precision, it amplifies the impact of superficial errors and highlights the need for stronger normalization or alternative evaluation schemes.

The zero-shot LLM-based approach introduces additional challenges related to controllability and reproducibility. Due to inherent non-determinism, identical prompts may yield different outputs across runs. Moreover, zero-shot extraction tends to over-generate terms or produce annotation-inconsistent spans, leading to reduced precision and limiting its reliability without post-processing.

From an ethical standpoint, terminology extraction from administrative documents may affect downstream applications in the public sector, such as document retrieval or decision-support systems. Extraction errors could propagate to such applications if not properly validated. In addition, the opacity of large language models raises concerns regarding transparency and accountability, particularly in settings where explainability is required.

These considerations underline the importance of human oversight and careful validation when applying automated term extraction in administrative domains. Future work should emphasize robustness, interpretability, and responsible deployment of language technologies in public-sector contexts.

6.4. Observed behavior

Compared to the supervised model, the zero-shot system tends to favor coverage over boundary precision: it often returns semantically plausible candidates but may over-generate and produce spans that are too long, especially in enumerations and long administrative sentences. This behavior aligns with the lower precision observed in evaluation.

7. Conclusion

This work addresses Subtask A of the ATE-IT shared task at EVALITA, focusing on automatic terminology extraction from Italian administrative documents in the waste-management domain. We compare a supervised BERT-based sequence labeling approach with a zero-shot large language model-based extraction pipeline.

Results show that the supervised BERT model achieves the highest term-level F_1 score, benefiting from explicit supervision and accurate boundary detection. The zero-shot approach, while less precise, demonstrates higher recall and the ability to identify relevant terminology without task-specific training.

Error analysis highlights challenges inherent to administrative texts, including long sentences, complex nominal structures, and strict exact-match evaluation. These factors affect both approaches in different ways and explain the observed performance differences.

Overall, this study provides a controlled comparison of supervised and zero-shot methods under the ATE-IT evaluation framework and clarifies their respective trade-offs in terms of precision, recall, and generalization for terminology extraction in administrative domains.

8. Declaration on Generative AI

During the preparation of this work, **Grammarly** was used for grammar and spelling checking. **ChatGPT** and **Gemini** were used for rephrasing and sentence polishing to improve clarity and readability. **ChatGPT** were also used to convert parts of the manuscript into $\text{\LaTeX}$ code and to adapt the document

to the CEUR-WS one-column CEUR-ART style (formatting assistance). In addition, **Google Translate** was used for text translation and subsequent language polishing. All content was reviewed and edited by the authors, who take full responsibility for the final manuscript.

References

- [1] K. Kageura, B. Umino, Methods of automatic term recognition: A review, *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.
- [2] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [3] F. Cutugno, A. Miaschi, A. P. Apro시오, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [4] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, *NAACL-HLT* (2019).
- [5] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. doi:10.1093/llc/fqad030.
- [6] K. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms: the C-value/NC-value method, *International Journal on Digital Libraries* 3 (2000) 115–130. doi:10.1007/s007999900023.
- [7] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, in: *International Conference on Learning Representations*, 2020.
- [8] L. Ouyang, J. Wu, X. Jiang, et al., Training language models to follow instructions with human feedback, in: *Advances in Neural Information Processing Systems*, 2022.
- [9] Google DeepMind, Gemini: A family of highly capable multimodal models, 2024. URL: <https://deepmind.google/technologies/gemini/>.
- [10] Meta AI, The llama 3.1 language model, 2024. URL: <https://ai.meta.com/llama/>.
- [11] N. Cirillo, et al., Ate-it: Automatic terminology extraction for italian, in: *Proceedings of EVALITA*, 2025. URL: <https://nicolacirillo.github.io/ate-it/>, shared Task description.
- [12] N. Cirillo, D. Vellutino, D. Nicoletti, E. Sabarese, B. Rubino, Itaist_gru, 2025. URL: <https://doi.org/10.5281/zenodo.15173712>. doi:10.5281/zenodo.15173712.
- [13] HuggingFace, Huggingface transformers, 2024. URL: <https://huggingface.co/transformers/>.

JTTE at ATE-IT: A CRF Model with Contextual Embeddings

Juliette Tonneau^{1,*}, Giorgio Maria Di Nunzio¹

¹Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

Automatic Term Extraction (ATE) is a core task in Natural Language Processing, aiming to identify domain-specific terminology that can support downstream applications such as information retrieval, ontology construction, and domain adaptation. This paper presents a sequence labeling approach to the ATE-IT shared task at EVALITA 2026, focusing on the extraction of terms from Italian institutional texts in the waste management domain. The task is formulated as a BIO tagging problem and addressed using a linear-chain Conditional Random Field (CRF) model. The proposed system integrates symbolic linguistic features obtained from spaCy, contextual semantic representations derived from a pretrained Italian BERT model, and weak domain knowledge through an automatically constructed glossary extracted from the training data. BERT embeddings are used as fixed contextual features and combined with token-level linguistic information within the CRF framework to enforce coherent span-level predictions. Experimental results on both development and test sets show that the proposed approach consistently improves Type F1-score over the baseline, with particularly notable gains in precision, indicating more accurate reconstruction of complete term boundaries. While recall remains lower than the baseline, the analysis highlights the effectiveness of combining contextual embeddings with structured decoding for Italian Automatic Term Extraction and outlines directions for improving generalization and recall in future work.

Keywords

Automatic Term Extraction, Sequence Labeling, Conditional Random Fields, BERT, Italian NLP

1. Introduction

Automatic Term Extraction (ATE) is a fundamental task in Natural Language Processing. Its goal is to identify domain-specific terms within a text [1, 2]. These terms can be multiword or single-word concepts that are relevant to a field. The terms extracted through ATE serve as foundation for multiple tasks such as information retrieval, machine translation, ontology construction, knowledge-graph enrichment, and domain adaptation of large language models (LLM) [3].

The Automatic Term Extraction - Italian Testbed (ATE-IT) [4] shared task is part of EVALITA 2026 [5]. It provides a space to perform ATE in the Italian Language and is centered around the field of waste management in institutional texts. In this domain, we can find single-word terms, synonyms, abbreviations, and multiword terms, which allows us to assess the efficiency of approaches.

The paper presents the solution for Subtask A: Term Extraction. The goal of this subtask is to extract terms relevant to the waste management domain from sentences drawn from a corpus related to the municipal waste management domain.

The remainder of this paper is organized as follows. Section 2 describes the proposed methodology, including problem formulation, linguistic preprocessing, domain glossary construction, feature extraction, and the CRF-based sequence labeling model. Section 3 presents the evaluation setup and experimental results on the development and test sets. Section 4 discusses the strengths and limitations of the system and outlines possible improvements. Finally, Section 5 concludes the paper and summarizes the main contributions of this work.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ julietteannelise.tonneau@studenti.unipd.it (J. Tonneau); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

ORCID 0000-0001-7116-9338 (G. M. Di Nunzio)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Description of the approach

2.1. Problem Formulation

The task addressed in this work is Automatic Term Extraction (ATE), formulated as a sequence labeling problem. Given a sentence drawn from the waste management domain, the objective is to identify all tokens that belong to domain-specific terms and to reconstruct complete term spans, which may consist of one or multiple consecutive tokens.

Formally, let X be a sequence of T tokens obtained after linguistic preprocessing of a sentence:

$$X = (x_1, \dots, x_i, \dots, x_T),$$

The goal is to predict a corresponding sequence of labels Y

$$Y = (y_1, \dots, y_i, \dots, y_T),$$

where each $y_i \in \{B - TERM, I - TERM, O\}$. The label B-TERM marks the beginning of a term, I-TERM marks the continuation of a term, and O denotes tokens that are not part of any term. A domain term is defined as any contiguous span of tokens starting with a B-TERM label and optionally followed by one or more I-TERM labels.

The task presents several challenges. First, term boundaries must be identified precisely, as partial matches are not considered correct under span-level evaluation. Second, labels are not independent: valid predictions must respect structural constraints imposed by the BIO scheme. For instance, an I-TERM label cannot occur without a preceding B-TERM. Finally, terms may have significant lexical and syntactic variability, requiring the model to integrate contextual, linguistic, and semantic information.

2.2. Overall Architecture

The system follows a pipeline architecture illustrated in Figure 1.

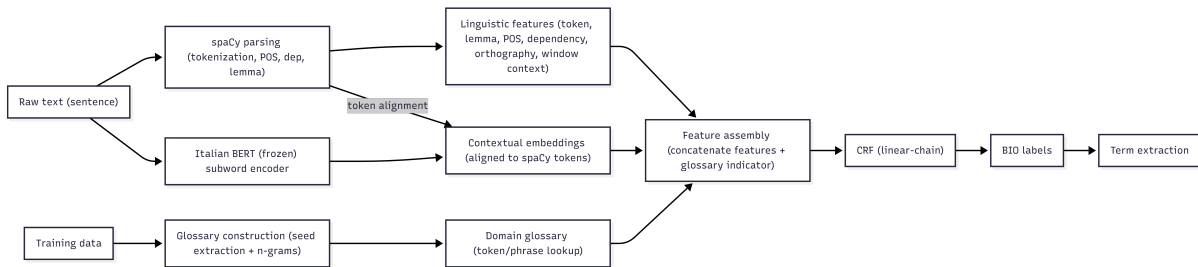


Figure 1: Overall architecture of the system

The main stages are the sentence parsing and linguistic annotations using spaCy, the automatic construction of a glossary from training data, the token-level feature extraction, including BERT-based contextual embeddings, the sequence labeling with a linear-chain Conditional Random Field, and the reconstruction of term spans from predicted BIO labels.

Each part of this pipeline will be described in the following sections.

2.3. Linguistic Preprocessing and Tokenization

All inputs are processed using a linguistic preprocessing pipeline based on the spaCy Italian language model `it_core_news_sm`. This step provides a structured representation of each sentence and establishes the tokenization scheme used consistently throughout training, validation, and inference.

2.3.1. Tokenization

Sentences are first segmented into tokens using spaCy’s rule-based tokenizer. Tokenization defines the fundamental units over which features are extracted and labels are predicted. It is necessary to maintain consistent token boundaries, as the next steps operate at the token level.

The spaCy tokenizer is applied uniformly in the `train`, `dev`, and `test` datasets, ensuring alignment between gold annotations, extracted features, and predicted labels. Special attention is given to preserving original token order and indices to support accurate reconstruction of multi-token terms during post-processing.

2.3.2. Linguistic Annotation

For each token, spaCy provides a set of linguistic annotations that are later exploited as symbolic features. These annotations include the following.

- `Lemma`, representing the canonical form of the token
- `Part-of-speech (POS) tags`, marking each word as a syntactic category
- `Morpho-syntactic tags`, adding a precision to the POS tags (eg.: verb tense, noun gender, and number)
- `Dependency relations`, describing the syntactic relations between the tokens
- `Orthographic attributes`, such as the presence of digits
- `Stopword indicators`, allowing the model to distinguish tokens that bear content from functional words

These annotations are stored in a structured token representation and will be used as features for the sequence labeling model.

2.3.3. Contextual Token Information

Beyond individual token attributes, contextual information is extracted by considering neighboring tokens within a fixed window. Concretely, for each token at position i in the spaCy Doc, we add the lowercased surface form and the POS tag of the previous token ($i - 1$) and of the next token ($i + 1$). These features are encoded as `prev_text`, `prev_pos`, `next_text`, and `next_pos`. When the current token is at the beginning or end of the sentence, the missing neighbor is handled using sentinel values: `<START>` for the previous token and `<END>` for the next token.

Features derived from preceding and following tokens encode local syntactic and lexical context, which is particularly important for identifying term boundaries.

2.4. Domain Glossary Construction

To incorporate domain-specific knowledge into the term extraction process, we automatically construct a domain glossary from the training data. The glossary is used as a weak semantic signal during feature extraction, without enforcing hard constraints on predictions.

2.4.1. Domain Seed Extraction

The glossary construction process begins with the identification of domain seed terms from the training corpus. All training sentences are first processed with spaCy to obtain token-level linguistic annotations. Candidate seed tokens are selected if the token is not a stopwords or punctuation mark, if the token belongs to a nominal or adjectival part-of-speech category, and if the token is not a temporal expression (e.g., `day`, `names`).

Candidate tokens are lemmatized and counted across the corpus. Seeds are selected by retaining the most frequent lemmas with a minimum frequency threshold of 5 occurrences. Among these, only the top 50 most frequent lemmas are kept. This procedure results in 50 domain seed terms, which are stored in an external file (`domain_seeds.txt`) and used as a binary token-level feature during CRF training.

2.4.2. N-Gram Extraction and Filtering

Using the extracted domain seeds, a glossary of candidate multi-word terms is constructed by extracting n-grams (up to a maximum length of six tokens) from the parsed training sentences. N-grams are filtered according to a set of syntactic and orthographic constraints to ensure linguistic validity. In particular, valid n-grams must contain no punctuation, begin and end with nominal or adjectival tokens, and include at least one noun or proper noun.

Additionally, only n-grams containing at least one previously identified domain seed are retained. This constraint ensures that extracted expressions are domain-relevant.

Filtered n-grams are then retained in the final glossary if they occur with sufficient frequency. The resulting glossary thus consists of domain-specific single-word and multi-word expressions that are both linguistically valid and frequent in the data.

The glossary is stored externally and is used during feature extraction to provide a binary indicator that signals whether a given token is present in the glossary.

2.5. Feature Extraction

Feature extraction is performed at the token level and aims to construct a unified representation that integrates symbolic linguistic hints, domain knowledge, and contextual semantic information. Linguistic, contextual window, and domain-related features are derived from spaCy annotations and the automatically constructed glossary, as described in the previous sections.

2.5.1. Contextual Semantic Features with BERT

We enrich each spaCy token with a contextual embedding extracted from a pretrained Italian BERT model (dbmdz/bert-base-italian-uncased). Because BERT uses WordPiece subword tokenization whereas the CRF operates on spaCy word-level tokens, we explicitly align subword representations to spaCy tokens.

Given a sentence tokenized by spaCy, we pass the list of spaCy token strings to the HuggingFace tokenizer with `is_split_into_words=True`. This ensures that the tokenizer preserves the mapping between each generated subword and its spaCy token. The sentence is then encoded by the BERT model (with frozen parameters), and we extract contextual representations from the last hidden layer (768 dimensions per subword).

To obtain one vector per spaCy token, we group all subword vectors that share the same word id (`encoded.word_ids()`) and average them (mean pooling). In the rare case where a spaCy token receives no aligned subwords (e.g., due to tokenizer edge cases), we assign a zero vector of the same dimensionality. The resulting 768-dimensional token embedding is concatenated with the symbolic token-level features and provided to the CRF as part of the emission representation. This alignment and mean-pooling strategy is implemented in our `BertEmbedder` module using the HuggingFace `transformers` library.

2.6. Sequence Labeling with Conditional Random Fields

To model dependencies between adjacent labels, the system employs a linear-chain Conditional Random Field (CRF) as the final prediction layer [6]. The model is implemented using the `sklearn_crfsuite` library, a Python interface to CRFsuite¹. The CRF operates on the token-level feature representations described in the previous section and produces a structured BIO label sequence for each input sentence.

2.6.1. Model Formulation

Let

¹<https://github.com/TeamHG-Memex/sklearn-crfsuite>

$$X = (x_1, x_2, \dots, x_T)$$

be a sentence of length T , where each x_t is the assembled feature representation of token t , and let

$$Y = (y_1, y_2, \dots, y_T)$$

be the corresponding sequence of BIO labels, with $y_t \in \{B - TERM, I - TERM, O\}$.

The CRF defines a conditional probability distribution over label sequences given the input features:

$$p(Y|X) = \frac{1}{Z(X)} \exp\left(\sum_{t=1}^T (\psi_{emit}(y_t, x_t) + \psi_{trans}(y_{t-1}, y_t))\right)$$

[7]

where:

- $\psi_{emit}(y_t, x_t)$ is the emission score, measuring the compatibility between label y_t and the feature vector of token t ,
- $\psi_{trans}(y_{t-1}, y_t)$ is the transition score, modeling dependencies between consecutive labels,
- $Z(X)$ is the partition function ensuring normalization.

Emission scores are linear functions of the token-level features, while transition scores are learned parameters capturing valid and frequent label transitions.

2.6.2. Training Objective

The model is trained by maximizing the conditional log-likelihood of the gold label sequences over the training data:

$$\mathcal{L} = \sum_{(X,Y)} \log p(Y|X)$$

Optimization is performed using L-BFGS, with both L1 and L2 regularization applied to diminish overfitting. Hyperparameters controlling regularization strength are selected via randomized search with cross-validation.

2.6.3. Inference

At inference time, the most probable label sequence is obtained by:

$$\hat{Y} = \arg \max_Y p(Y|X)$$

which is computed using the Viterbi algorithm. This decoding ensures that predictions respect BIO constraints, such as preventing invalid transitions

2.6.4. Advantages of CRF-Based Decoding

The CRF explicitly models label dependencies and optimizes predictions at the sequence level. This is particularly important for term extraction, where multi-token expressions must be labeled consistently and boundaries must be accurately identified.

By combining rich token-level features and structured decoding, the CRF enforces global coherence in the predicted label sentences.

3. Results and Evaluation

3.1. Evaluation Setup

The system is evaluated on both a development set and a held-out test set using the same preprocessing, feature extraction, and decoding pipeline. Models are trained exclusively on the training data, with hyperparameter selection performed via cross-validation on the training set. The development set is used for validation and qualitative analysis, while the test set is reserved for final performance reporting. This evaluation protocol ensures a clear separation between model selection and final assessment.

The baseline provided by the ATE-IT shared task organizers is a zero-shot large language model approach based on Gemini 2.5 Flash. In this setup, the model is prompted to extract domain-specific terms directly from the input sentences without any task-specific fine-tuning or supervised training. Sentences are processed in batches of 20, and the model outputs are post-processed to obtain the predicted term spans.

3.2. Evaluation Metrics

Performance is evaluated using instance-based and type-based metrics.

Instance-based evaluation (Micro precision, recall, and F1-score) considers each predicted term occurrence as a separate instance. A predicted term instance is counted as correct only if its span exactly matches a gold-standard term instance in the same sentence. Precision is computed as the proportion of correctly predicted term instances among all predicted instances, while recall corresponds to the proportion of gold term instances that are correctly identified. Micro-averaging aggregates true positives, false positives, and false negatives across all sentences before computing the final score.

Type-based evaluation (Type precision, recall, and F1-score) instead operates on the set of unique terms extracted over the corpus. In this setting, multiple occurrences of the same term are counted only once. A predicted term type is considered correct if it exactly matches a gold term type. Precision measures the proportion of correctly predicted unique terms among all predicted unique terms, while recall measures the proportion of gold unique terms that are successfully extracted.

Both evaluations require exact span matching, but they differ in whether frequency of occurrence is taken into account (instance-based) or not (type-based).

3.3. Overall Results

Model	Precision	Recall	F1-score
<i>Micro (Token-level)</i>			
Baseline	0.439	0.616	0.513
Proposed system	0.717	0.674	0.695
<i>Type (Span-level)</i>			
Baseline	0.372	0.636	0.470
Proposed system	0.638	0.583	0.609

Table 1

Performance comparison on the development set between the baseline and the proposed system using token-level (Micro) and span-level (Type) evaluation metrics.

Table 1 reports the performance of the baseline and the proposed system on the development set. The proposed system outperforms the baseline at the token level, achieving higher precision, recall, and F1-score. At the span level, it yields higher precision and F1-score, while the baseline attains higher recall. Overall, the results indicate a clear improvement in extraction accuracy and boundary coherence on the development set.

Table 2 reports the performance of the baseline system and the proposed approach on the test set. At the token level, the baseline achieves the highest recall and F1-score, while the proposed system attains higher precision. At the span level, the proposed system outperforms the baseline in both precision

Model	Precision	Recall	F1-score
<i>Micro (Token-level)</i>			
Baseline	0.497	0.559	0.526
Proposed system	0.555	0.448	0.496
<i>Type (Span-level)</i>			
Baseline	0.435	0.508	0.469
Proposed system	0.561	0.447	0.498

Table 2

Performance comparison on the test set between the baseline and the proposed system using token-level (Micro) and span-level (Type) evaluation metrics.

and F1-score, indicating more accurate reconstruction of complete term spans, whereas the baseline achieves higher recall.

In general, better results are obtained on the development set. This is a common bias in machine learning. Implicitly, the model is adapted to the development set. Indeed, hyperparameters, architectural choices, and feature engineering decisions are selected based on the performance of the model on this set.

4. Discussion

This section analyzes the behavior of the proposed system in the light of the results of the previous section.

4.1. Strengths of the system

A first notable strength of the proposed system is its consistent improvement in span-level (Type) F1-score over the baseline on both the development and test sets. On the development set, the proposed system achieves a higher Type F1-score (0.609 vs. 0.470), and this improvement is maintained on the test set (0.498 vs. 0.469). This indicates that the system is more effective at reconstructing complete term spans with correct boundaries, which is the primary objective of domain-specific term extraction.

At the token level, the system demonstrates higher precision than the baseline on both datasets. On the development set, Micro precision increases from 0.439 to 0.717, while on the test set it increases from 0.497 to 0.555. This suggests that the proposed approach produces fewer false positive labels, reflecting more selective and confident predictions. The combination of contextual BERT embeddings with linguistic features appears to help the model better distinguish term and non-term tokens.

The use of CRF-based structured decoding also contributes positively to output quality. Across both datasets, predictions show coherent BIO label sequences, and extracted terms correspond to contiguous, well-formed spans. This structural consistency is reflected in the improved span-level F1-scores and confirms the benefit of modeling label dependencies explicitly rather than relying on independent token classification.

4.2. Limitations

Despite these improvements, several limitations are apparent from the results.

First, the proposed system exhibits lower recall compared to the baseline at both the token and span levels. On the test set, Micro recall decreases from 0.559 to 0.448 and at the span level, it drops from 0.508 to 0.447. This indicates that while the system is more precise, it tends to miss a portion of relevant terms, especially less noticeable or ambiguous ones.

Second, the difference between development and test performance highlights a degree of generalization difficulty. While gains on the development set are important, improvements on the test set are more moderate, particularly in terms of F1-score. This suggests that some of the learned patterns, such

as those induced by the domain glossary or contextual features, may be sensitive to the distribution of the training and development data.

Finally, although dependency labels and contextual embeddings are used, the model still relies on token-level feature representations. Complex syntactic constructions or long-range dependencies may not be fully captured, which can contribute to boundary errors or missed multi-word expressions.

4.3. Possible improvements

Several paths could be explored to address these limitations.

A first one would be to improve boundary detection by adding different features to the model, such as head information, or explicit noun-phrase boundaries. These features would help the CRF model to decide where spans should start and end.

Improving generalization should be a priority in future improvements. The gap between development and test results suggests a slight overfitting on the data. Improving semantic generalization of BERT features by adding a contrastive embedding regularization and/or using a multi-layer BERT feature mixing could considerably help in this direction.

After those last tasks, another way to upgrade the model would be to implement a two-stage decoding.

In the current system, the CRF performs sequence labeling over all tokens in a sentence, implicitly considering every possible span as a potential term. While this allows for flexible predictions, it also leads to a large search space and encourages conservative behavior, as the model must balance precision and recall across many unlikely span configurations. A two-stage approach addresses this issue by restricting the CRF's decision space to a set of plausible candidate spans. First, the system generates a set of candidate term spans for each sentence. The CRF would then be applied only within the boundaries of candidate spans. By operating within a reduced and more relevant search space, the CRF can make more confident predictions, particularly for multi-token expressions.

5. Conclusion

This paper presented a sequence-labeling approach to Automatic Term Extraction for the ATE-IT shared task at EVALITA 2026, focusing on the waste management domain in Italian. The proposed system combines linguistic preprocessing with contextual semantic representations derived from a pretrained Italian BERT model and leverages a linear-chain Conditional Random Field to enforce structured BIO decoding. Domain knowledge is incorporated through the automatic construction of a glossary extracted from the training data, used as a weak semantic signal during feature extraction.

Experimental results on both development and test sets show that the proposed system consistently improves span-level (Type) F1-score over the baseline, indicating more accurate reconstruction of complete term boundaries. In particular, gains in precision demonstrate the effectiveness of combining contextual embeddings with symbolic linguistic features and structured decoding. However, the analysis also highlights current limitations in generalization and recall.

Overall, the proposed approach provides a strong and extensible baseline for Italian Automatic Term Extraction and offers several promising directions for further improvement within the ATE-IT framework.

Declaration on Generative AI

During the preparation of this work, the author(s) used X-GPT-5 in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] K. Kageura, B. Umino, Methods of automatic term recognition: A review, *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.
- [2] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. doi:10.1093/11c/fqad030.
- [3] Y. Chun, M. Kim, D. Kim, C. Park, H. Lim, Enhancing Automatic Term Extraction with Large Language Models via Syntactic Retrieval, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 9916–9926. doi:10.18653/v1/2025.findings-acl.516.
- [4] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [5] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [6] N. Nguyen, Y. Guo, Comparisons of sequence labeling algorithms and extensions, in: *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, Association for Computing Machinery, New York, NY, USA, 2007, pp. 681–688. doi:10.1145/1273496.1273582.
- [7] C. Sutton, A. McCallum, An introduction to conditional random fields, 2010. URL: <https://arxiv.org/abs/1011.4088>. arXiv:1011.4088.

MKTE at ATE-IT: CRF-Based Term Extraction for Italian Waste Management Documents

Minseok Kim^{1,*}, Giorgio Maria Di Nunzio¹

¹Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

This paper presents a Conditional Random Fields (CRF) sequence-labelling system for sentence-level Automatic Terminology Extraction in the ATE-IT shared task at EVALITA 2026, targeting Italian municipal waste-management documents. The system models term spans using linguistically informed features, including POS tags, lemmas, dependency relations, and morphological patterns, and applies task-specific post-processing to satisfy lowercasing, sentence-level de-duplication, and non-nested output constraints. On the official test set, our submission achieves micro-F1 scores of 0.519 and 0.529, corresponding to ranks 5 and 3 across submitted runs, and yields higher precision than the baseline. However, recall remains comparatively lower, and we observe a substantial gap between training and test performance, suggesting overfitting and limited generalization to unseen documents. We also find only modest gains over a zero-shot large language model approach, indicating that large pre-trained models may encode substantial implicit terminology knowledge even without task-specific supervision. We conclude by outlining directions for improving robustness, including richer semantic representations, data augmentation, and ensemble strategies to reduce the generalization gap.

Keywords

Term Extraction, Conditional Random Fields, Italian NLP, Waste Management

1. Introduction

Automatic Terminology Extraction (ATE) aims to identify and extract domain-specific terms from specialised corpora, with an emphasis on lexical units (single- and multi-word expressions) that denote domain concepts rather than unique real-world referents [1, 2, 3]. Unlike general named entity recognition, ATE focuses on discovering technical terminology that constitutes the conceptual backbone of a domain and is therefore closely connected to downstream tasks such as indexing, knowledge organisation, and domain adaptation.

The ATE-IT task [4] at EVALITA 2026 [5] targets term extraction from Italian municipal waste-management documents, a setting that combines institutional register with domain terminology, abbreviations, and complex multi-word administrative expressions. In Subtask A, systems receive sentences with metadata (e.g., document/paragraph/sentence identifiers) and must extract domain terms (including nouns, verbs, and adjectives) under explicit output constraints: terms must be lowercased, de-duplicated at sentence level, and non-nested. Performance is measured with two complementary metrics: micro-averaged F1 computed over sentence-level instances and type F1 computed over unique term types across the dataset [6].

We model term extraction as a sequence labelling problem using Conditional Random Fields (CRF), which provide a principled framework for predicting structured label sequences while integrating rich, hand-engineered linguistic features [7]. Our system combines token- and context-level cues derived from POS tags, dependency relations, lemmas, and morphology to learn patterns that distinguish domain terms from common words and to improve multi-word boundary detection. We report results on the official test set and submitted runs, and we analyse error profiles to characterise the strengths and limitations of a feature-based CRF approach under ATE-IT's sentence-level constraints.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26–27, Bari, IT

*Corresponding author.

✉ minseok.kim@studenti.unipd.it (M. Kim); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

ORCID 0000-0001-7116-9338 (G. M. Di Nunzio)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Description of the System

Our pipeline consists of four stages: BIO conversion, feature extraction, CRF modeling, and post-processing.

2.1. BIO Tagging and Preprocessing

We convert the training data to sequence labeling format using the BIO (Begin-Inside-Outside) tagging scheme, which is a standard representation for sequence labeling tasks in natural language processing. Each token in a sentence receives one of three labels: B (Begin) marks the first token of a term, I (Inside) marks continuation tokens in multi-word terms, and O (Outside) marks tokens that are not part of any term. This representation allows the model to capture both single-word terms (marked with a single B label) and multi-word terms (marked with B followed by one or more I labels), while also maintaining clear boundaries between adjacent terms.

For each sentence in the training data, we first tokenize the text using SpaCy's Italian tokenizer, which handles Italian-specific linguistic phenomena such as apostrophes, contractions, and punctuation. Then, for each gold-standard term provided in the annotations, we perform case-insensitive substring matching to locate the term within the original sentence text. The matching process operates at the character level to handle potential discrepancies between term boundaries and tokenization boundaries.

When a match is found, we examine which tokens in the SpaCy tokenization overlap with the character span of the term. The first token that overlaps with the term receives the B label, and all subsequent tokens that fall within the term span receive I labels. All tokens that do not overlap with any gold term receive the O label. This character-based approach ensures robust alignment even when the gold term boundaries do not perfectly align with token boundaries, which can occur due to differences in tokenization strategies or the presence of punctuation within terms.

After processing all gold terms for a sentence, we obtain parallel sequences of tokens and BIO labels. These paired sequences constitute the training examples for the CRF model, where each sequence represents a complete sentence context that the model can use to learn patterns of term occurrence and boundary detection.

2.2. Feature Extraction

We extract features using SpaCy's `it_core_news_lg` model, which provides tokenization, lemmatization, POS tagging, and dependency parsing. For each token, we extract:

Lexical features: lowercase form, boolean indicators for uppercase, title case, and digits, plus 2-character and 3-character prefixes and suffixes helping capture patterns.

Morpho-syntactic features: POS tag indicating grammatical category and dependency relation showing syntactic function, which help distinguish technical terms from functional words.

Lexical-semantic features: lemma providing the dictionary form for generalization, and stop word indicator identifying common functional words.

Orthographic features: token shape abstracting the pattern (e.g., "Xxxxx" for title case, "dddd" for digits).

Contextual features: lowercase form, POS tag, and lemma from the previous and next tokens, enabling the model to learn sequential patterns. Sentence boundary tokens receive special BOS/EOS markers.

Bias feature: constant value allowing the model to learn label priors.

This feature set balances information with efficiency, providing linguistic context, while maintaining reasonable dimension.

2.3. CRF Model

We employ a linear-chain Conditional Random Field that predicts label sequences conditioned on inputs. Unlike local classifiers, CRFs model dependencies between adjacent labels, ensuring coherent BIO

sequences. We use `sklearn-crfsuite` with L-BFGS optimization, L1 and L2 regularization coefficients of 0.1, maximum 200 iterations, and all possible transitions enabled (to learn state transition constraints).

For training, we combine the provided training and development datasets. The CRF learns feature weights maximizing the conditional probability of correct label sequences. We validated our approach using 10-fold cross-validation on the combined data.

2.4. Prediction and Post-processing

For prediction on new sentences, we apply the same feature extraction pipeline used during training. Each sentence is tokenized using `SpaCy`, and for each token, we extract the same comprehensive feature dictionary described in Section 2.2. This produces a sequence of feature dictionaries representing the sentence.

The trained CRF model then predicts the most likely BIO label sequence for the sentence using the Viterbi algorithm, which efficiently computes the highest-probability path through the label sequence space by using dynamic programming. The Viterbi algorithm takes into account both the emission probabilities (how likely each label is for each token given its features) and the transition probabilities (how likely each label is to follow the previous label), finding the globally optimal label sequence rather than making independent per-token decisions.

After obtaining the predicted BIO label sequence, we reconstruct the extracted terms through a simple scanning process. We iterate through the tokens and their predicted labels from left to right. When we encounter a B label, we start a new term and add the corresponding token text to it. Any following I labels extend the current term by appending their token text (with spaces between tokens). When we encounter an O label or reach the end of the sentence, we finalize the current term and add it to the list of extracted terms for that sentence.

Finally, we apply post-processing steps to ensure strict compliance with the task requirements specified in the ATE-IT guidelines. All extracted terms are converted to lowercase, as required by the task specification. This normalization step ensures consistency regardless of the original capitalization in the source text. Within each sentence, we remove duplicate terms, keeping only unique term strings. If the same term is extracted multiple times from the same sentence (which can occur if it appears in multiple locations), only one instance is retained in the output. The extracted terms are formatted as a JSON structure with the required fields: `document_id`, `paragraph_id`, `sentence_id` (copied from the input), and `term_list` (containing the list of unique, lowercased terms extracted from that sentence).

This post-processing stage ensures that our system output exactly matches the format expected by the evaluation system and satisfies all task constraints, including the prohibition against nested terms (which is implicitly handled by the BIO scheme, as each token can belong to at most one term).

2.5. Implementation

The complete system is implemented in Python 3. The main dependencies are: `SpaCy` 3.7.2 with the `it_core_news_lg-3.7.2` model for Italian natural language processing, providing tokenization, lemmatization, POS tagging, dependency parsing, and morphological analysis. `sklearn-crfsuite` 0.3.6 for CRF modeling. `scikit-learn` 1.5.1 for cross-validation infrastructure and evaluation utilities. `pandas` 2.2.2 for structured data manipulation and result aggregation. `tqdm` 4.66.2 for progress bars during data processing and model training.

The system architecture follows a modular design with clear separation of concerns. Separate Python modules handle data loading and parsing, BIO conversion, feature extraction, model training, prediction, and evaluation. This modularity facilitates experimentation with different feature sets, model configurations, and post-processing strategies. The codebase is organized to make it straightforward to modify individual components (such as adding new features or changing the CRF hyperparameters) without affecting other parts of the pipeline.

Table 1
Official test set results

System	Rank	Micro-P	Micro-R	Micro-F1
Baseline (Gemini)	–	0.497	0.559	0.526
Our CRF (Run on 1)	5	0.569	0.476	0.519
Our CRF (Run on 2)	3	0.654	0.444	0.529

Table 2
10-fold cross-validation on training data

Metric	Mean	Std Dev
Micro-Precision	0.799	0.023
Micro-Recall	0.698	0.031
Micro-F1	0.745	0.019
Type-Precision	0.761	0.028
Type-Recall	0.628	0.034
Type-F1	0.687	0.022

3. Results

We submitted a run with minor variations in configuration. Table 1 shows official test results compared to the Gemini-based baseline.

Run on 2 achieved micro-F1 of 0.529 (rank 3), slightly outperforming the baseline (0.526). Run on 1 obtained 0.519 (rank 5), performing comparably to baseline. Both runs show substantially higher precision than baseline (0.569 and 0.654 vs. 0.497) but lower recall (0.476 and 0.444 vs. 0.559). Run 2’s higher precision (0.654) comes at the cost of lower recall (0.444), indicating more conservative predictions.

Table 2 shows cross-validation results on training data.

Cross-validation results (micro-F1: 0.745) significantly exceed test performance (0.519-0.529), that probably indicates overfitting to training patterns or systematic differences between training and test distributions.

4. Discussion

4.1. Performance Analysis

Our results reveal several key patterns. Both runs favor precision over recall, producing more conservative predictions than baseline. This suggests our feature-based CRF learned to be selective, identifying high-confidence terms while missing borderline cases. The baseline’s higher recall may indicate the will to extract more candidates, though with more false positives.

The large gap between cross-validation (0.745) and test performance (0.519-0.529) is concerning. Several factors may contribute: the test set may contain vocabulary or terminology patterns underrepresented in training data. Our features may have captured training correlations that don’t generalize. The training corpus size may limit learning really robust patterns. Municipal waste management terminology varies across regions and administrative contexts, and our model may lack exposure to more test-set terminology.

Our marginal improvement over the zero-shot Gemini baseline, despite using supervised learning with annotated data, suggests large language models possess substantial implicit knowledge about technical terminology. However, the baseline’s lower precision indicates it extracts too many spurious terms, while our CRF’s higher precision shows it learned more discriminative patterns.

4.2. Error Analysis and Limitations

Several factors likely limit recall performance:

Rare terms: Many domain terms appear infrequently in training data. CRFs require multiple examples to learn reliable patterns. Rare terms, especially those with unusual structures may not be extracted consistently.

Multi-word complexity: Complex noun phrases like "impianto di trattamento rifiuti" require learning both syntactic patterns and domain relevance. If similar structures appear with non-term content in training, generalization will probably fail.

Domain coverage: Waste management includes multiple sub-domains: regulatory language, waste categories, collection procedures, and infrastructure. Unbalanced training coverage probably leads to uneven performance across the content types.

Context-dependence: Some words are domain terms only in specific contexts. For example, "conferire" (deliver) becomes technical in "conferire rifiuti" (deliver waste). Training examples may not clearly demonstrate such distinctions sometimes.

Feature limitations: Our features capture morpho-syntactic patterns but lack semantic representations. Word embeddings or contextual representations from pre-trained language models might provide richer semantic information.

Nested terms: The task prohibits nested terms, yet related terms may overlap. The BIO scheme must choose one representation, and our model may not resolve such cases consistently with gold annotations.

4.3. What Worked

Despite limitations, several aspects proved effective. Rich linguistic features enabled learning discriminative patterns, with POS tags particularly helpful in constraining predictions to appropriate word classes. The CRF's structured prediction ensures coherent BIO sequences, reducing invalid transitions and improving boundary detection. High precision means extracted terms are generally correct, valuable for applications where false positives are costly.

4.4. Future Improvements

Several directions could improve performance:

Enhanced semantics: Incorporating embeddings or contextualized representations from Italian language models (BERT, RoBERTa) could provide richer semantic information, helping with rare terms and context-dependent terminology.

Ensemble methods: Combining our CRF with neural models, rule-based extractors, or LLM outputs could leverage complementary strengths, balancing precision and recall.

Domain adaptation: Fine-tuning pre-trained models on unlabeled waste management texts or incorporating external resources like glossaries could provide additional supervision.

5. Conclusion

We presented a CRF-based approach to term extraction for Italian waste management documents, achieving competitive performance with micro-F1 scores of 0.519 and 0.529 (ranks 5 and 3). While demonstrating higher precision than baseline, lower recall and substantial train-test performance gaps highlight probable overfitting and generalization challenges. The modest improvement over zero-shot language models suggests that large pre-trained models possess considerable implicit terminology knowledge. Future work should focus on enhanced semantic representations, data augmentation, and ensemble methods to bridge the generalization gap.

Declaration on Generative AI

The author have employed Generative AI tools in the preparation of this work. All code development, experimentation, analysis, and writing were performed by the author with assistance from large language models or other AI generation systems. The author also used Claude 4.5 and ChatGPT to check grammar and spelling. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] K. Kageura, B. Umino, Methods of automatic term recognition: A review, *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.
- [2] K. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms: the C-value/NC-value method, *International Journal on Digital Libraries* 3 (2000) 115–130. doi:10.1007/s0079999900023.
- [3] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. doi:10.1093/11c/fqad030.
- [4] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [5] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [6] R. Verborgh, M. Vander Sande, P. Colpaert, S. Coppens, E. Mannens, R. Van de Walle, Web-scale querying through linked open data, *Semantic Web* 9 (2018) 49–69.
- [7] J. D. Lafferty, A. McCallum, F. C. N. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: *Proceedings of the 18th International Conference on Machine Learning (ICML)*, 2001, pp. 282–289.

OATE at ATE-IT: A Hybrid Approach for Automatic Terminology Extraction in Italian

Omar Arab^{1,*}, Giorgio Maria Di Nunzio¹

¹Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

This paper describes a hybrid system developed for the ATE-IT shared task at EVALITA 2026. The system combines supervised sequence labeling based on a BERT+CRF architecture with semantic post-processing driven by term embeddings. Graph-based ranking methods were also explored but performed poorly since was not suited because of the nature of the dataset. Experimental analysis shows that embedding-based semantic similarity, enhanced with large language model (LLM)-generated domain knowledge, represents the dominant signal for terminology filtering in this setting.

Keywords

Automatic Terminology Extraction, EVALITA, Italian, BERT, Conditional Random Fields, Semantic Embeddings

1. Introduction

Automatic Terminology Extraction (ATE) is a specialized Natural Language Processing task aimed at identifying and extracting domain-specific terms from textual corpora [1, 2]. Despite its apparent simplicity, ATE poses several challenges, including the detection of multi-word expressions, ambiguous term boundaries, and the distinction between domain-specific terms and generic noun phrases.

In this context, recent ATE approaches increasingly frame term identification as a span-detection problem, borrowing sequence-labelling architectures from Named Entity Recognition while retaining classical insights about unithood (whether a candidate forms a stable multi-word unit) and termhood (whether it is domain-specific) [3, 4]. Hybrid pipelines typically combine contextual encoders with linguistically informed post-processing to enforce task constraints (e.g., de-duplication, normalization, and nested-term removal) and to better control boundary errors on multi-word expressions, which remain a primary source of false positives and fragmented predictions in morphologically rich languages such as Italian.

The ATE-IT [5] shared task at EVALITA 2026 [6] focuses on Italian terminology extraction, with particular attention to technical domains such as waste management and environmental regulation. Compared to English, Italian presents additional challenges due to its rich morphology and complex syntactic structures. These characteristics require models capable of capturing both local structural constraints and global semantic context.

2. System Architecture and Methodology

The proposed system follows a hybrid pipeline composed of three main stages: (i) high-recall candidate extraction via supervised sequence labeling, (ii) semantic enrichment of candidate terms, and (iii) embedding-based ranking and filtering. While graph-based ranking techniques were initially investigated, they were ultimately found to be ineffective for the given dataset and therefore play only a marginal role in the final system.

ATE EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26–27, Bari, Italy

*Corresponding author.

✉ omar.arab@studenti.unipd.it (O. Arab); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

🆔 0000-0001-7116-9338 (G. M. Di Nunzio)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2.1. Preprocessing and Tokenization

The corpus undergoes a structured preprocessing pipeline before model training. First, all documents are normalized to UTF-8 encoding to avoid inconsistencies caused by special characters. Non-textual artifacts (e.g., HTML remnants or malformed symbols) are removed. Excess whitespace is normalized, and line breaks are standardized. No lowercasing is applied since we rely on the cased version of the Italian BERT model to preserve morphological and orthographic distinctions relevant for domain terminology.

Tokenization is performed using the WordPiece algorithm implemented in the `dbmdz/bert-base-italian-cased` model. WordPiece decomposes rare or morphologically complex words into subword units. For example, a compound technical term may be split into multiple sub-tokens prefixed with `##`.

To ensure correct supervision during training, we adopt an explicit alignment strategy between original tokens and subword units. Labels are assigned only to the first subword of each original token, while subsequent subwords are masked during loss computation. Specifically:

- If a token is labeled as B-TERM, its first subword receives B-TERM and the remaining subwords are assigned a special ignore index.
- If labeled as I-TERM, the same rule applies.
- Non-term tokens are labeled as O.

Character-level offset mappings provided by the tokenizer are used to reconstruct spans precisely after inference. This guarantees that extracted candidate terms preserve their exact surface form in the original documents.

2.2. Neural Sequence Labeling with BERT and CRF

Terminology extraction is formulated as a supervised sequence labeling task using the BIO tagging scheme. The BIO tagging scheme encodes span boundaries by assigning a label to each token in the sequence. In this framework, B-TERM marks the beginning of a terminology span, I-TERM marks tokens that continue a previously started term, and O marks tokens that do not belong to any terminology expression. This representation allows the model to capture multi-word domain expressions while explicitly preserving boundary information. By distinguishing between span-initial and span-internal tokens, the BIO formalism provides structural constraints that are particularly important for accurately modeling complex and morphologically rich terminology.

A pre-trained Italian BERT model (`dbmdz/bert-base-italian-cased`) is used to generate contextualized representations for each token.

To enforce global consistency in label predictions, a Conditional Random Field (CRF) layer is applied on top of BERT emissions [7]. While a token-level classifier may produce locally optimal predictions, it cannot model dependencies between adjacent labels. The CRF explicitly learns transition constraints, preventing invalid sequences such as an I-TERM tag following an O tag.

Empirically, the CRF layer significantly improved span-level consistency, particularly for multi-word terms.

2.3. Training Strategy

The model is trained using the AdamW optimizer [8] with discriminative learning rates. A lower learning rate is applied to the BERT encoder to preserve pre-trained linguistic knowledge and to avoid catastrophic forgetting, while higher rates are used for the CRF layer to speed up learning. This strategy allows effective adaptation to the terminology extraction task. In our experiments we have kept 5 epochs since this number has shown better performances in terms of loss.

2.4. Candidate Reconstruction

Predicted BIO sequences are converted into candidate terms by extracting maximal contiguous spans of B-TERM and I-TERM labels. To avoid reconstruction artifacts introduced by subword tokenization, term spans are mapped back to the original text using character-level offsets. This guarantees that extracted terms preserve the exact surface form found in the source documents.

2.5. Graph-Based Ranking: Empirical Limitations

We experimented with graph-based ranking methods, including PageRank-style centrality and TextRank, to estimate candidate-term salience from co-occurrence and (where available) syntactic relations [9, 10].

Each candidate term is represented as a node, and edges connect terms that are observed as related within a fixed context window and/or within the same dependency neighbourhood. Edge weights correspond to the strength of association, operationalised here as (weighted) co-occurrence frequency. Once the graph is constructed, we compute node centrality scores and use them as an additional signal in post-processing (e.g., re-ranking or filtering low-salience candidates).

Empirically, however, this signal had limited impact on the ATE-IT dataset: documents are relatively short and specialised terms exhibit limited repetition, leading to sparse, weakly connected graphs. As a consequence, centrality tended to privilege generic, high-frequency expressions that act as hubs across many contexts, rather than genuinely domain-specific terminology.

This behaviour is consistent with known failure modes of frequency-driven graph centrality when domain terms have low recurrence; addressing it would likely require stronger association measures, domain-adaptive priors, or document level aggregation, which we leave for future work.

2.6. Semantic Augmentation via Large Language Models

To enhance domain coverage, we leveraged a large language model accessed via the Groq API. The model employed was `llama-3.1-8b-instant`, accessed in November 2025. The model was prompted in zero-shot mode to extract domain-specific terminology from Italian environmental legislation, in particular *D.Lgs. 152/2006*. The prompting instructions were written in Italian to maximize domain and linguistic alignment.

The model was instructed to return only technical terms relevant to waste management and environmental regulation, one per line, without additional explanations. The extracted terminology was aggregated across document chunks and subsequently deduplicated and filtered to remove noise.

While LLM-generated candidates cannot be considered fully reliable due to possible variability across runs, they provide broader conceptual coverage and introduce domain signals that may not be sufficiently represented in the supervised training data. Rather than being used directly as predictions, these terms serve as a semantic reference set in the embedding validation phase.

2.7. Embedding-Based Semantic Ranking

The dominant ranking signal in the final system is based on semantic similarity in an embedding space. Each extracted candidate term is represented as a dense embedding vector, together with a set of domain-relevant reference terms obtained from model extractions and LLM-generated terminology.

For each candidate term, cosine similarity is computed against all reference terms in the semantic space. A candidate is retained if at least one reference term exceeds a predefined similarity threshold τ . This nearest-neighbor semantic validation strategy allows the system to identify candidates that are semantically aligned with the domain, even when they do not exhibit strong frequency or co-occurrence signals.

In practice, this approach proved effective at filtering generic expressions while preserving domain-specific terminology, particularly in sparse corpora where traditional graph-based or frequency-based signals are unreliable.

2.8. Final Ranking Strategy

The final ranking relies primarily on embedding-based semantic similarity, with extraction confidence from the BERT+CRF model used as a secondary signal. Graph-based centrality was evaluated but its effect on the final scoring was practically nil.

3. Results

The system was evaluated using the official ATE-IT evaluation script, reporting both micro-averaged and type-based Precision, Recall, and F1-score. Micro-level metrics measure performance over all term occurrences, while type-level evaluation considers unique term instances, providing complementary perspectives on extraction quality.

The OA-TE system (run 2) achieved a micro Precision of 0.581, a micro Recall of 0.522, and a micro F1-score of 0.550, outperforming the official baseline (micro F1 = 0.526). At the type level, OA-TE obtained a Precision of 0.569, a Recall of 0.492, and a type F1-score of 0.528, again showing a clear improvement over the baseline (type F1 = 0.469). The observed performance indicates a clear precision-recall trade-off introduced by the semantic filtering stage. While the embedding-based validation slightly reduces recall by discarding candidates that are weakly aligned with the target domain, it contributes to a modest but consistent gain in precision.

Although graph-based centrality was experimentally integrated into the scoring function, it did not influence the final evaluation metrics. The observed Precision, Recall, and F1-scores remained unchanged within rounding tolerance across tested configurations. Consequently, the final submitted system does not depend on PageRank-based refinement, and the reported results correspond to the embedding-driven filtering strategy.

4. Discussion

The experimental findings show how strongly Automatic Terminology Extraction methods depend on the characteristics of the dataset. In the ATE-IT task, documents are relatively short and many domain-specific terms appear only a few times. This has important consequences for the effectiveness of different ranking strategies. Methods that rely mainly on frequency or global connectivity signals naturally struggle when repetition is limited.

Graph-based ranking techniques such as PageRank were explored as a complementary signal to supervised extraction. The idea was that terms that frequently co-occur with other relevant terms would become more central in a co-occurrence graph and therefore more representative of the domain. We built weighted graphs based on sentence-level co-occurrence and also enriched them using syntactic proximity information obtained from dependency parsing. However, due to the sparsity of the dataset, the resulting graph was weakly connected and centrality scores were not very discriminative. In practice, PageRank tended to favor terms that were structurally well connected rather than truly domain-specific. When integrated into a hybrid scoring function together with embedding similarity, the graph signal did not produce consistent improvements in Precision, Recall, or F1-score. This suggests that in small and specialized corpora, graph connectivity alone may not be sufficient to model domain relevance.

The supervised extraction component based on BERT with a CRF decoder proved much more reliable. The CRF layer helps enforce valid transitions within the BIO tagging scheme and reduces inconsistent label sequences. This is especially important for Italian, where technical terms often consist of multi-word expressions with morphological agreement. By modeling dependencies between adjacent labels, the CRF improves boundary consistency and reduces span fragmentation. Nevertheless, the sequence labeling approach also has limitations. The BIO scheme cannot represent nested or overlapping terminology, and the model remains dependent on the coverage and quality of the annotated training data.

The embedding-based semantic validation stage was the most impactful part of the system. By representing candidate terms in a contextual embedding space and comparing them to a domain-specific reference set, the model filters candidates based on semantic consistency rather than frequency. This approach is particularly effective in sparse datasets, where repetition is not a reliable indicator of termhood. Embedding similarity allows the system to retain rare but semantically aligned terms while discarding generic noun phrases that are not truly domain-specific.

At the same time, this strategy introduces some sensitivity. The similarity threshold directly affects the precision–recall trade-off. A lower threshold increases recall but may allow more noise, while a higher threshold improves precision but risks filtering out valid terms. Although the threshold was tuned empirically, it remains a manually chosen hyperparameter and may not generalize perfectly to different domains.

Another important aspect concerns the use of LLM-derived terminology for semantic augmentation. While the prompting strategy was carefully designed to restrict outputs to relevant technical terms, large language models can show variability across runs and may introduce subtle bias. The inclusion of externally generated terms strengthens domain coverage, but it also introduces dependence on model behavior at a specific time and version. A more integrated solution, such as domain-adaptive fine-tuning instead of post-hoc filtering, could potentially provide more stable improvements.

More generally, the current pipeline treats extraction and semantic validation as two separate stages. While this modular design simplifies experimentation, it may limit the interaction between structural boundary detection and semantic alignment. Future work could explore joint optimization strategies or contrastive training objectives that directly encourage domain-consistent representations during fine-tuning.

Overall, the experiments suggest that for the ATE-IT dataset, semantic alignment in contextual embedding space is more effective than graph-based structural connectivity. In sparse and highly specialized corpora, embedding similarity provides a stronger and more stable signal for domain consistency. However, the system remains sensitive to threshold selection, training data coverage, and external semantic augmentation, indicating room for further methodological refinement.

Declaration on Generative AI

During the preparation of this work, generative AI tools were used to assist with language refinement. All generated content was reviewed and edited by the author, who takes full responsibility for the final manuscript.

References

- [1] K. Kageura, B. Umino, A survey of automatic term extraction, *Terminology* 16 (2010) 259–289.
- [2] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. doi:10.1093/dsh/11c/fqad030.
- [3] K. Kageura, B. Umino, Methods of automatic term recognition: A review, *Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication* 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.
- [4] K. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms: the C-value/NC-value method, *International Journal on Digital Libraries* 3 (2000) 115–130. doi:10.1007/s007999900023.
- [5] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.

- [6] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [7] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, Proceedings of the 18th International Conference on Machine Learning (2001) 282–289.
- [8] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations (ICLR), 2019.
- [9] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking : Bringing order to the web, in: The Web Conference, 1999. URL: <https://api.semanticscholar.org/CorpusID:1508503>.
- [10] R. Mihalcea, P. Tarau, TextRank: Bringing Order into Text, in: D. Lin, D. Wu (Eds.), Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 404–411.
- [11] X. Wan, J. Xiao, Singlerank: A graph-based keyword extraction method, Proceedings of the 23rd International Conference on Computational Linguistics (COLING) (2010) 465–473.
- [12] T. B. Brown, B. Mann, N. Ryder, et al., Language models are few-shot learners, Advances in Neural Information Processing Systems 33 (2020).

Peacemaker at ATE-IT: Automatic term extraction from Italian text for waste management data using encoder model

Mahdi Bakhtiyarzadeh^{1,*}, Hadi Bayrami Asl Tekanlou^{2,*} and Jafar Razmara^{1,*}

¹Department of Computer Science, University of Tabriz, 29 Bahman Boulevard, Tabriz 51666-16471, Iran

²Department of Computer Science, University of Tabriz, 29 Bahman Boulevard, Tabriz 51666-16471, Iran

Abstract

The development of automatic term extraction has become increasingly important in modern technology. Automatic term extraction can be found in virtually every search engine that is currently available to users. Recent advancements have provided promising results for the extraction of automatic terms; however, accurate labeling is difficult because of several factors, such as the limited number of annotated documents available for training and the complexity of extracting multi-word expressions due to shifts in the domain. In this paper, we will present a low-cost and interpretable method of automatic term extraction, developed specifically for Task A of the ATE Shared Task. This new method utilizes fine-tuning extraction strategies that can run on a small amount of computational resources. We evaluated our automated system using both type-level and micro-level measures of precision, recall, and F1-score to measure both complementary aspects of the extraction performance. According to the experimental results, our proposed approach achieves consistent and balanced performance compared to other teams. Even though the technique itself is relatively straightforward, it serves as a good starting point for low-resource models. Overall, the findings point toward the possibility of significant future advancements (in model expansion) with higher-level performance still able to retain their ability to be interpreted.

Keywords

Automatic Term Extraction, Large Language Models, Ontology Learning, Domain-Specific Terminology, Information Extraction

1. Introduction

Terminology is the beating heart of sentences, especially in tasks such as Automatic Term Extraction (ATE), where identifying domain-specific concepts from corpora is crucial for applications like machine translation, knowledge extraction, and semantic analysis [1, 2]. The "Automatic Term Extraction" (ATE) focuses on tasks such as corpus construction, unithood (measuring word co-occurrence), termhood (assessing domain relevance), and recognizing "variants" or how a term can be expressed differently from the original way [3]. Recent research utilizing large language models for few-shot ATE demonstrated considerable progress when operated under conditions where limited resources exist. Additionally [1], systematic reviews have documented ongoing development of methods for extracting multi-word terms from specific types of text through Transformer's architecture and ability to extract terms across multiple languages and domains [2, 4]. These advances play a significant role in the field of computer-assisted translation (CAT), where having bilingual terms available improves the quality of machine translations [5] as well as supporting the evaluation of term extraction software against gold standard datasets such as ACLRD-TEC2.0 and MAGMATic [6, 7]. Monolingual and Multilingual ATE [8], extending into specialized fields such as biomedical literature [9] and media bias analysis [10], is supported by datasets created from comparable corpora. It is common practice when improving ATE to

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ m.bakhtiyarzadeh1403@ms.tabrizu.ac.ir (M. Bakhtiyarzadeh); h.bayrami1403@ms.tabrizu.ac.ir (H. B. A. Tekanlou); razmara@tabrizu.ac.ir (J. Razmara)

🌐 <https://github.com/Mahdi8424> (M. Bakhtiyarzadeh); <https://github.com/HadiBayrami/> (H. B. A. Tekanlou)

🆔 0009-0002-7206-6116 (H. B. A. Tekanlou); 0000-0002-6320-8517 (J. Razmara)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

integrate terminological resources [11] with well defined robust evaluation frameworks [12]. However, challenges still exist regarding the collection of semi-automated data for Systematic Reviews [13].

This study makes the following contributions:

- **Robust Data Alignment and Preparation:** We implement a custom preprocessing pipeline featuring token-level semantic alignment (BIOtagging), specifically tailored for the Italian language and the ATE-IT task. This process accurately maps and converts multiword term instances from the raw gold standard into the correct BTERM, ITERM, O sequence labels, resolving token boundary ambiguities essential for effective Token Classification.
- **High-Fidelity Feature Extraction Architecture:** We utilize a high-performance, context-aware Transformer-based Token Classification architecture, leveraging the pre-trained dbmdz/bert-base-italian-cased model. This provides deep, contextual semantic representations optimized for token-level predictions within the Italian regulatory domain.
- **Environment Stability and Reproducibility:** We ensure the stability and reproducibility of the fine-tuning process by explicitly pinning critical Python dependencies (including transformers and accelerate) to verified, compatible versions. This guarantees API compatibility across varying runtime environments, mitigating common failure modes and allowing for reliable training execution.

2. Related Work

As presented in the previous section, Automatic Term Extraction (ATE) is critical for extracting domain-specific concepts that underpin the core tasks of natural language processing and knowledge management. The initial strategies that dealt with ATE were based on the combination of linguistic and statistical techniques to address issues such as unithood (the occurrence of words together) and termhood (the relevance of terms to domains). Examples of hybrid methods used on specialized corpora, such as in computer science and medicine, include the combination of statistical measures (mutual information and C-value/NC-value) with shallow linguistic criteria (part of speech) [14]. Various approaches have shown strong performance across domains, with similarity scores closely matching those for terms represented via traditional word-based representations in document similarity evaluations. However, the new methods will need to compensate for cases where multi-word terms have nesting problems [14]. Additionally, the comparative evaluations of term recognition systems indicate that combining several different approaches, such as using voting methods, resulted in better performance than using any one method alone when evaluated against corpus data from Wikipedia text, but performance varied from domain to domain and was not as high when using data from the Genia corpus in the life sciences [15]. Tools based on shallow grammars designed for use with noun-phrases in technical circles and combined with the application of clustering to accommodate variations in their structure (e.g., using variation of insertion structure) have generated very high levels of precision ranging from 93% to 98% when constructing an automated thesaurus [16]. PAT-tree approach also has been shown to enable the construction of adaptive keyword extraction systems, reducing dependency on rigid lexicons and word segmentation of documents written in Asian languages, such as Chinese, and facilitating their effective retrieval or classification when using these systems [17]. Multilingual and Monolingual ATE advancements take advantage of comparable corpora to develop data sets that can be used to support variant recognition and cross-lingual transfer. One particular contribution is the development of a data set specifically for both monolingual and multilingual ATE that allows terms to be extracted from comparable sources and resolve resource scarcity in low-resource languages [8, 16]. This reflects an interest in leveraging corpus-based semantic connections to create associations to MWEs by way of Information Extraction, and using these associations to compare the MWEs to thesauri within the Agriculture and Medicine domains [18]. For Agriculture, rule-based systems like RelExOnt have automated vocabulary extraction as well as relation discovery, providing ontologies with an 86.89% accuracy rate in the creation process [19]. The combination of statistical, linguistic, and hybrid extractors

has increased the accuracy of Spanish medical text extraction. It demonstrates the value of ensemble methods for different extraction methods [12]. Recent advancements in medical text extraction systems incorporating complete terminological resources and comprehensive evaluation methods allow for greater reliability of ATE systems (Automated Term Extraction). Incorporating external lexicons specific to the domain and other validated sources of information into automation models for extraction provide significant increases in recognizing multifaceted terms [18].

3. Task Description and Dataset

3.1. Task

This section describes the parameters for the ATE-IT Shared Task 2026 and its components, as well as the dataset used in the evaluation. It also describes the system architecture created for use in the evaluation; our contribution was in **Subtask A: Term Extraction**.

3.1.1. ATE-IT Shared Task 2026: Subtasks and Primary Focus

The ATE-IT Shared Task 2026 [20] has been organized around an evaluation of Automated Term Extraction (ATE) in a large scale way at the specific domain level of waste management in Italy. The challenge is divided into two distinct subtasks of increasing complexity:

1. **Subtask A: Term Extraction.** The main goal in the task of Term Extraction is to identify and extract individual domain-specific terms from the corpus sentences.
2. **Subtask B: Term Variants Clustering.** The primary objective of Term Variants Clustering is to group extracted terms that refer to the same underlying concept. This requires semantic and morphological analysis of terms so that synonymy and lexical variations can be handled.

In this study, we will be participating only in Subtask A: Term Extraction. Our main goal will be to process the sentences of the specialized corpus accurately and identify the municipal waste management domain-specific terms.

3.2. Corpus Description

Ate-it challenge organizers supplied the official textual source for the ATE-IT shared task, which contained textual examples related directly to the field of municipal waste management (an area of environmental concern). The challenge also provided guidelines regarding how to use the corpus dataset for this challenge, in order to develop models based on the provided training data and evaluate models against a pre-defined testing dataset in accordance with the established structure of the challenge. Tables 1 and 2 present an overview of the ATE-IT data statistics and representative examples from the dataset, respectively.

Table 1

ATE-IT data statistic overview.

Dataset	Samples
Train	2308
Development	577
Test	1142

Table 2

Representative examples from the dataset

sentence_text	term_list
CONSIDERATO che la situazione in cui si trova attualmente il Comune è riconducibile all'ipotesi contemplata nelle previsioni di cui al citato art. 191, in quanto sussistono gravi condizioni e fondate ragioni di tutela della salute pubblica e dell'ambiente, che risulterebbero inevitabilmente pregiudicate in caso di mancato ricorso temporaneo ad una speciale forma di gestione del servizio di raccolta per i rifiuti provenienti da luoghi adibiti ad uso di civile abitazione in cui alloggino persone risultate positive al Covid-19, in quarantena obbligatoria ai sensi dell'articolo 1 lettera e del D.P.C.M. 8 marzo 2020	"gestione del servizio di raccolta per i rifiuti", "servizio di raccolta per i rifiuti provenienti da luoghi adibiti ad uso di civile abitazione"
Al fine di consentire la raccolta dei rifiuti nei contenitori interni agli stabili, il proprietario singolo o i condomini, in solido fra loro, hanno l'obbligo di esporre gli stessi nei giorni e nelle ore stabiliti sul tratto viario prospiciente l'immobile di competenza e di riporli all'interno dei cortili o delle pertinenze condominiali, dopo l'avvenuto servizio di raccolta.	"esporre", "raccolta dei rifiuti", "servizio di raccolta"
La richiesta di attivazione del servizio deve essere presentata dall'utente al gestore dell'attività di gestione tariffe e rapporto con gli utenti entro novanta (90) giorni solari dalla data di inizio del possesso o della detenzione dell'immobile, a mezzo posta, via e-mail o - 5 mediante sportello fisico e online di cui all'Articolo 19, compilando l'apposito modulo scaricabile dalla home page del sito internet del gestore in modalità anche stampabile, disponibile presso gli sportelli fisici, laddove presenti, ovvero compilabile online art.	"gestore", "tariffe", "utente"

4. Proposed Approach

The system that has been developed to complete ATE_IT Subtask A (extraction of Italian vocabulary) uses a Sequence Labeling Framework. The identification of terms has been changed into classifying every single word in a sentence. We used a fine-tuned state-of-the-art transformer model for the extraction of Vocabulary.

4.1. Data Preprocessing and Alignment

A critical component of the methodology is the custom preprocessing pipeline, which converts raw, sentence-level term annotations into a format consumable by the sequence labeling model.

- **BIO Tagging Scheme:** The standard B-I-O (Begin, Inside, Outside) tagging scheme is employed: B-TERM for the first token of a term, I-TERM for subsequent tokens, and O for non-term tokens.
- **Token-Term Alignment:** Term-to-token correspondence is ensured by using offset mappings provided by the tokenizer to create a mapping between character-level terms and the subword tokens from the model. To avoid double labels and overlap between terms that are themselves "overlapping," a mask is created for each character in the input sentence as it emerges from the model for each term. The mask makes sure that for every character position, it can only be mapped to one term, thus ensuring that a single character can't be allocated to two separate terms.
- **Label Encoding:** Symbolic BIO labels are converted into numerical IDs (O=0, B-TERM=1, I-TERM=2). Special tokens (e.g., [CLS], [SEP]) and padded tokens are masked with a label ID of -100, which is ignored by the loss function.

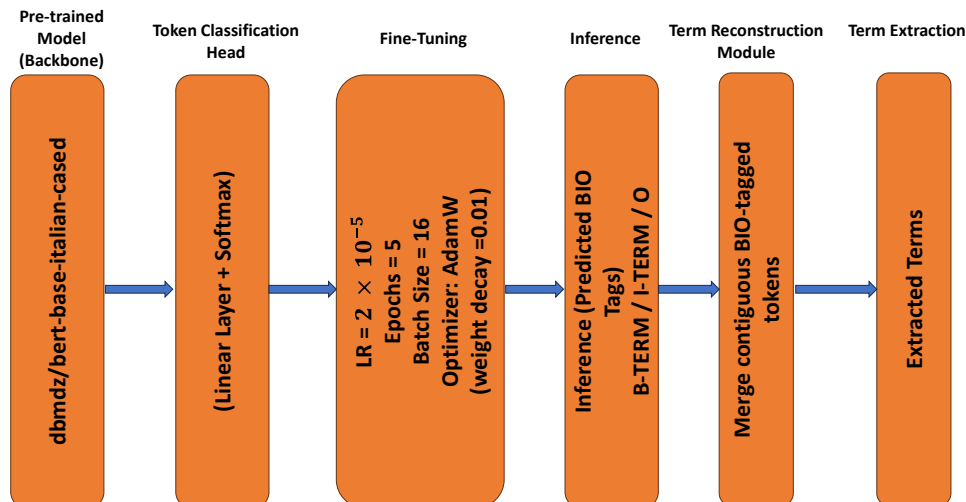


Figure 1: Overview of the system architecture used for token-level term extraction.

4.2. Modeling and Training

The core of the system is the *dbmdz/bert-base-italian-cased* model, a cased **BERT (Bidirectional Encoder Representations from Transformers)** variant pre-trained on an extensive Italian corpus. This model is adapted for **Token Classification** by appending a linear layer for transfer learning. The system architecture is illustrated in Figure 1.

5. Results

In this section, we describe the results obtained in the Automatic Term Extraction 2026 shared task. A total of 9 teams participated in the competition. The present paper presents results based on micro and type based evaluation metrics. Micro measures consider the total count of each type of term (e.g., phone numbers, names) that appear in the data set and provide a measure of accuracy of the entire extraction process. Type measures provide insight into how well the system is able to find different types of terms within the dataset.

5.1. Baselines

As a reference point, Table 3 presents the baseline results for both type-based and micro-based evaluation metrics. The following is our first baseline, which should be used to compare all participating systems. This baseline was created using the latest version (gemini-2.5-flash) of a closed-source large language model, in a zero-shot mode. The input text was processed in 20 instance batches according to the predefined prompt.

Table 3

Baseline model results using type-based and micro-based evaluation metrics

Type Precision	Type Recall	Type F1	Micro Precision	Micro Recall	Micro F1
0.435	0.508	0.469	0.497	0.559	0.526

5.2. Evaluation Metrics

The criteria used to evaluate the submitted work included a comparison between the automated term extraction and the corresponding gold standard annotations (as previously outlined). At the same time, type-level and micro-level metrics provide separate initial views of system performance. Type-level metrics provide an aggregate measure of how many unique domain terms have been identified by the system regardless of their frequency of occurrence; micro-level metrics provide detailed performance data on a sentence-by-sentence basis by accumulating all "True," "False Positive," and "False negative" results from all sentences to arrive at totals. Additionally, the precision, recall and F1 score calculations were derived from both type-level and micro-level evaluations allowing for assessment between extracted term correctness and completeness from both perspectives.

Type Metrics

- **TP = number of extracted unique terms that appear in the gold set**
- **FP = number of extracted unique terms that do not appear in the gold set**
- **FN = number of gold unique terms that were not extracted**

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

Micro Metrics [21]

- **TP_s = number of terms extracted from sentence s that match the gold standard**
- **FP_s = number of terms extracted from sentence s that do not match the gold standard**
- **FN_s = number of gold standard terms in sentence s that were not extracted**

$$Precision_{\text{micro}} = \frac{\sum_{s \in D} TP_s}{\sum_{s \in D} (TP_s + FP_s)}$$

$$Recall_{\text{micro}} = \frac{\sum_{s \in D} TP_s}{\sum_{s \in D} (TP_s + FN_s)}$$

$$F1_{\text{micro}} = \frac{2 \cdot Precision_{\text{micro}} \cdot Recall_{\text{micro}}}{Precision_{\text{micro}} + Recall_{\text{micro}}}$$

5.3. Team Results

The outcomes from our application of the method we've described will now be discussed in detail. The performance of our system is assessed with the metrics shown in Section 5.2, and as for how our performance measures up against the results reported by each of the other teams that participated in this study, it is important to first compare these results so that we can accurately identify where the submitted systems rank with respect to one another and any associated strengths/weaknesses. For reference, Table 4 presents the detailed type-level and micro-level evaluation metrics achieved by the Peacemaker team, which are used as a baseline comparison in our analysis. In order to compare our model's performance against all evaluated dimensions, we provide a multi-dimensional view of our model's performance along with those of its competitors through the use of a radar plot (Figure 2), which allows us to quickly compare Precision, Recall and F1-score evaluations for the two evaluation types (type-level and micro-level) evaluated.

Table 4

Peacemaker team results using type-based and micro-based evaluation metrics

Type Precision	Type Recall	Type F1	Micro Precision	Micro Recall	Micro F1
0.430	0.455	0.442	0.497	0.476	0.486

6. Discussion

The results of the proposed method provide valid insight into how lightweight and low-engineering techniques behave in the domain of ontology term extraction even though it is not producing highest performance level. Evidence to support this claim is captured in the radar plot, which displays balanced precision and recall sections between the type-level and micro-level metrics. Therefore, this suggests that the proposed method produces stable extraction as opposed to maximally optimizing precision scores through repetitive extraction cycles.

While our experimental results show that the proposed method ranks lower than others, we have observed that for both the Type and Micro levels the Precision and Recall values are similar, indicating that this method does not rely heavily on just using high-frequency terms. We conclude that our proposed method has consistent extraction performance for both frequent and less frequent terms from many different evaluation granularities. Thus, our proposed method provides a strong baseline for future researchers in this area with a reasonable level of computational and implementation complexity.

It should be noted that our system was created using limited resources for both computation and modeling, which prevented us from being able to take full advantage of many of the larger or more

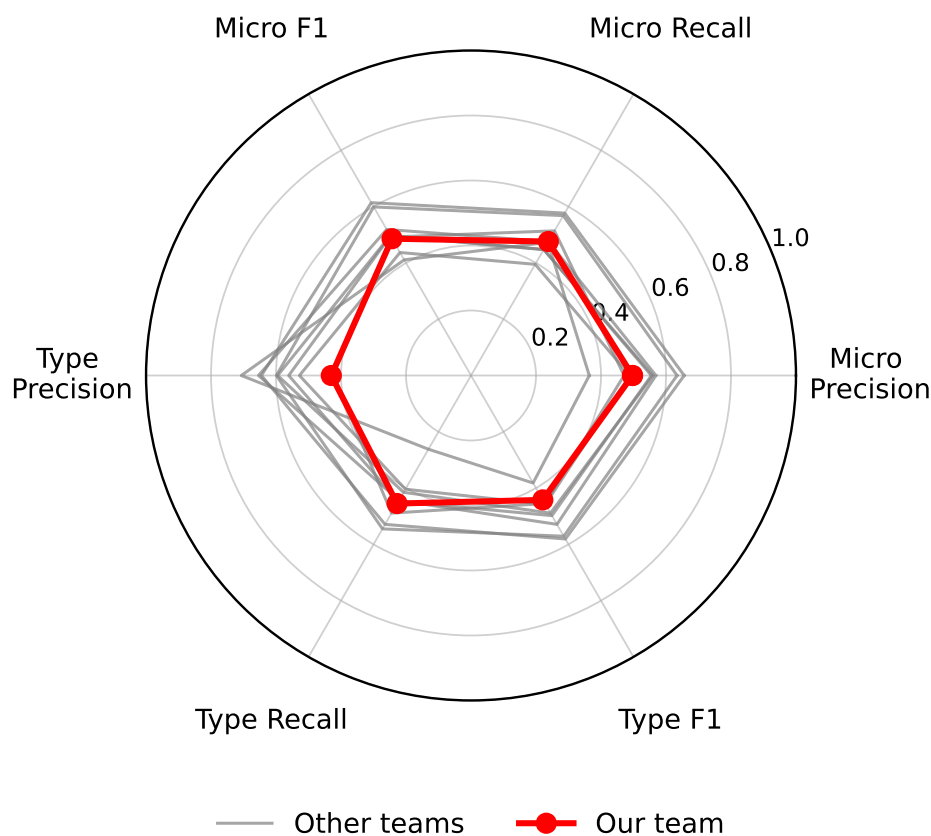


Figure 2: Radar plot comparing the performance of our approach with other participating systems across type-level and micro-level Precision, Recall, and F1-score.

specialized models that are available to some teams that have a higher ranking. If the teams were able to use better models and were able to increase their computational capabilities, it is likely that their overall performance—especially Recall—would also improve greatly. However, the current results show that the proposed method is effective and competitive and has the advantage of being lightweight.

7. Conclusions

In this research, we proposed an easy-to-understand, lightweight process for extracting ontology terms using data captured during the shared task evaluation phase and applying type-level and micro-level metrics to evaluate the performance of our solution. Our reduced resources and choice of more constrained models did not hinder our application from demonstrating steady and well-balanced extraction results across all models and evaluations, which implies that the extraction capabilities of our modeling process are not based solely on selecting terms that appear very frequently. The fact that our team came in 7th place out of the 9 teams that competed shows how well we stack up against other teams, even though we had far fewer resources than most of the other teams that took part in the competition; therefore, our system remains highly competitive based on what we have demonstrated.

Declaration on Generative AI

Generative AI tools played a supplemental and shallow role in our work. We used generative AI for brainstorming and developing ideas on how to tackle the Automatic Term Extraction (ATE) task; they were also helpful for enhancing the clarity and organization of notes and drafts. However, we did not rely on generative AI to create any part of the dataset, including the annotations, experimental results, or final models. All authors made the decisions about method choice, implementation, and analysis. After using these tools/services, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] S. Banerjee, B. R. Chakravarthi, J. P. McCrae, Large language models for few-shot automatic term extraction, in: A. Rapp, L. Di Caro, F. Meziane, V. Sugumaran (Eds.), *Natural Language Processing and Information Systems*, Springer Nature Switzerland, Cham, 2024, pp. 137–150.
- [2] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of automatic term extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. URL: <https://doi.org/10.1093/llc/fqad030>. doi:10.1093/llc/fqad030.
- [3] K. Heylen, D. D. Hertog, Automatic term extraction, 2014. URL: <https://api.semanticscholar.org/CorpusID:184348418>.
- [4] C. Lang, L. Wachowiak, B. Heinisch, D. Gromann, Transforming term extraction: Transformer-based approaches to multilingual term extraction across domains, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Association for Computational Linguistics, Online, 2021, pp. 3607–3620. URL: <https://aclanthology.org/2021.findings-acl.316/>. doi:10.18653/v1/2021.findings-acl.316.
- [5] M. Arcan, M. Turchi, S. Topelli, P. Buitelaar, Enhancing statistical machine translation with bilingual terminology in a CAT environment, in: Y. Al-Onaizan, M. Simard (Eds.), *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track*, Association for Machine Translation in the Americas, Vancouver, Canada, 2014, pp. 54–68. URL: <https://aclanthology.org/2014.amta-researchers.5/>.
- [6] B. QasemiZadeh, A.-K. Schumann, The ACL RD-TEC 2.0: A language resource for evaluating term extraction and entity recognition methods, in: N. Calzolari, K. Choukri, T. Declerck, S. Goggi, M. Grobelnik, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis (Eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, European

- Language Resources Association (ELRA), Portorož, Slovenia, 2016, pp. 1862–1868. URL: <https://aclanthology.org/L16-1294/>.
- [7] R. Scansani, L. Bentivogli, S. Bernardini, A. Ferraresi, MAGMATiC: A multi-domain academic gold standard with manual annotation of terminology for machine translation evaluation, in: M. Forcada, A. Way, B. Haddow, R. Sennrich (Eds.), *Proceedings of Machine Translation Summit XVII: Research Track*, European Association for Machine Translation, Dublin, Ireland, 2019, pp. 78–86. URL: <https://aclanthology.org/W19-6608/>.
 - [8] A. Rigouts Terryn, V. Hoste, E. Lefever, In no uncertain terms: a dataset for monolingual and multilingual automatic term extraction from comparable corpora, *Language Resources and Evaluation* 54 (2020) 385–418. URL: <https://doi.org/10.1007/s10579-019-09453-9>. doi:10.1007/s10579-019-09453-9.
 - [9] P. Tiwari, S. Upreti, S. Dehdashti, M. S. Hossain, TermInformer: unsupervised term mining and analysis in biomedical literature, *Neural Comput. Appl.* 37 (2020) 1–14.
 - [10] T. Spinde, L. Lin, S. Hinterreiter, I. Echizen, Leveraging large language models for automated definition extraction with taxomatic a case study on media bias, 2025. URL: <https://arxiv.org/abs/2504.00343>. arXiv:2504.00343.
 - [11] S. Aubin, T. Hamon, Improving term extraction with terminological resources, 2006. URL: <https://arxiv.org/abs/cs/0609019>. arXiv:cs/0609019.
 - [12] J. Vivaldi, H. Rodríguez, Evaluation of terms and term extraction systems: A practical approach, *Terminology* 13 (2007) 225–248. doi:10.1075/term.13.2.06viv.
 - [13] L. Schmidt, A. N. Finnerty Mutlu, R. Elmore, B. K. Olorisade, J. Thomas, J. P. T. Higgins, Data extraction methods for systematic review (semi)automation: Update of a living systematic review, *F1000Res.* 10 (2021) 401.
 - [14] E. E. Milios, Y. Zhang, B. He, L. Dong, Automatic term extraction and document similarity in special text corpora, 2003. URL: <https://api.semanticscholar.org/CorpusID:3128871>.
 - [15] Z. Zhang, J. Iria, C. Brewster, F. Ciravegna, A comparative evaluation of term recognition algorithms, in: N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, D. Tapias (Eds.), *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, European Language Resources Association (ELRA), Marrakech, Morocco, 2008. URL: <https://aclanthology.org/L08-1281/>.
 - [16] D. Bourigault, C. Jacquemin, Term extraction + term clustering: an integrated platform for computer-aided terminology, in: *Proceedings of the Ninth Conference on European Chapter of the Association for Computational Linguistics, EACL '99*, Association for Computational Linguistics, USA, 1999, p. 15–22. URL: <https://doi.org/10.3115/977035.977039>. doi:10.3115/977035.977039.
 - [17] L.-F. Chien, Pat-tree-based keyword extraction for chinese information retrieval, *SIGIR Forum* 31 (1997) 50–58. URL: <https://doi.org/10.1145/278459.258534>. doi:10.1145/278459.258534.
 - [18] E. Morin, C. Jacquemin, Projecting corpus-based semantic links on a thesaurus, in: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99*, Association for Computational Linguistics, USA, 1999, p. 389–396. URL: <https://doi.org/10.3115/1034678.1034739>. doi:10.3115/1034678.1034739.
 - [19] N. Kaushik, N. Chatterjee, Automatic relationship extraction from agricultural text for ontology construction, *Information Processing in Agriculture* 5 (2018) 60–73. URL: <https://www.sciencedirect.com/science/article/pii/S2214317317300227>. doi:<https://doi.org/10.1016/j.inpa.2017.11.003>.
 - [20] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
 - [21] M. Röder, R. Usbeck, A.-C. N. Ngomo, Gerbil – benchmarking named entity recognition and linking consistently, *Semantic Web* 9 (2018) 605–625. URL: <https://journals.sagepub.com/doi/abs/10.3233/SW-170286>. doi:10.3233/SW-170286. arXiv:<https://journals.sagepub.com/doi/pdf/10.3233/SW-170286>.

SMTE at ATE-IT: Ensemble Term Extraction with Italian BERT, spaCy, and Vocabulary-Based Filtering

Sofia Maule^{1,*}, Giorgio Maria Di Nunzio¹

¹Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

This paper describes the SMTE system developed for the shared task ATE-IT (Subtask A: Automatic Term Extraction) at EVALITA 2026 in Italian institutional texts in the municipal waste-management domain. Given a sentence, the goal is to extract domain-specific single- and multi-word terms while complying with task constraints such as lowercasing, sentence-level de-duplication, and the removal of nested terms. SMTE follows an ensemble approach that combines (i) a fine-tuned Italian BERT token-classification model trained with BIO tags to detect term spans, (ii) a trained spaCy sequence labeller to provide complementary candidates, and (iii) a vocabulary built from gold training terms to support normalization and filtering. A final post-processing stage cleans and canonicalises predictions, removes truncated spans and nested terms, and formats the output JSON for evaluation and submission. We report our official evaluation results and discuss strengths, typical errors, and directions for future improvements.

Keywords

Automatic Term Extraction, Terminology, Italian NLP, Domain-specific term extraction, BERT token classification, spaCy, ATE-IT, EVALITA 2026

1. Introduction

Automatic Term Extraction (ATE) is a core task in Natural Language Processing (NLP) and terminology work. Given a domain-specific corpus, its goal is to identify the terms that denote key concepts in that domain, including single and multi-word expressions (e.g., technical noun phrases, derived terms, and common domain abbreviations) [1]. Unlike Named Entity Recognition (NER) [2], which focuses on the mentions of named entities such as people, organizations, or places, ATE targets domain terminology that may not correspond to proper names and is aimed primarily at building terminological resources for downstream applications such as information retrieval, ontology construction, knowledge graph enrichment, and domain adaptation of language models.

EVALITA 2026 is a periodic evaluation campaign for Italian NLP and speech technologies across different tasks [3]. The ATE-IT shared task provides a large-scale benchmark for ATE in Italian and offers a realistic setting based on institutional texts from the municipal waste-management domain. This domain introduces challenges such as lexical variation, abbreviations, synonyms, and complex multiword terms, making it a useful testbed for robust terminology extraction methods [4].

In this work, we specifically tackle the ATE-IT Subtask A (Term Extraction) [4]. In this task, systems receive a collection of sentences and must output, for each sentence, a list of domain-relevant terms. The task imposes strict output constraints: extracted terms must be lowercased, duplicates within the same sentence are not allowed, and nested terms must be avoided (e.g., if a longer term is extracted, its inner subspans should not be returned unless they occur independently). Performance is measured using two complementary metrics: Micro F1, which evaluates precision and recall across all sentences and occurrences [5], and Type F1, calculated based on unique extracted term types. In this context, we present our approach, named SMTE, which is an ensemble system that combines neural sequence labeling and resource-based filtering to extract accurate term spans under ATE-IT constraints.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ sofia.maule@studenti.unipd.it (S. Maule); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

🆔 0000-0001-7116-9338 (G. M. Di Nunzio)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

All code, notebooks, and scripts used to reproduce the experiments and generate the submission files are publicly available at https://github.com/sofiamauale/ATE-IT_SofiaMaule.

The remainder of this paper is organized as follows: Section 2 describes the proposed SMTE system and its main components; Section 3 reports experimental results; and Section 4 discusses strengths, limitations, and future directions.

2. Description of the system

SMTE is an ensemble pipeline for ATE-IT Subtask A that extracts domain-relevant terms from each sentence while enforcing the shared-task output constraints (lowercase, sentence-level de-duplication, and no nested terms) [4]. The final submission system combines two complementary sequence labellers: a BERT-based token classifier and a spaCy-based tagger, and merges their candidates through a vocabulary derived from the gold training terms. A final cleaning stage normalizes and canonicalises the extracted spans, removes noisy or truncated candidates, applies de-duplication and nested-term removal, and formats the output JSON for official evaluation and submission.

All components of SMTE were developed starting from the baseline code and reference implementations released during the course tutorship for the ATE-IT task, which provided working pipelines and multiple system variants. We then adapted and extended this material to build our final submission system [6].

2.1. Preprocessing and data preparation

All sentences are normalized before training and inference to reduce noise and comply with the shared-task constraints. In particular, we:

- replace non-breaking spaces with regular spaces,
- collapse multiple whitespace characters,
- normalize apostrophes by mapping typographic quotes to a single form,
- lowercase the text (coherently with the uncased BERT checkpoint and with the required output format), and
- remove non-printable control characters.

The same normalization is applied to the gold terms used for training.

The input data are loaded from the official ATE-IT JSON files (sentence-level records with a `term_list`) and mapped to the internal representation used to train the sequence labelers..

2.2. BERT token-classification extractor

The main extractor is a token-classification model based on the Italian checkpoint `dbmdz/bert-base-italian-uncased` [7]. We fine-tuned it to detect term spans using a BIO tagging scheme with three labels: O, B-TERM, and I-TERM [8].

Training labels are generated automatically from the gold term lists by locating each term occurrence in the sentence string and mapping the resulting character-level spans to BERT subword tokens using offset mappings. To reduce spurious matches, a candidate match is accepted only if it satisfies a simple word-boundary check, i.e., the characters immediately before and after the match are either absent or non-alphanumeric.

For the final run submitted, the model is fine-tuned in the union of the official training and development sets using the HuggingFace Trainer [9]. The final training configuration is as follows:

- learning rate 2×10^{-5} ;
- batch size 16;
- 7 epochs;
- weight decay 0.01;

- linear learning-rate schedule with warmup ratio 0.1;
- random seed 42.

Mixed-precision training (FP16) is enabled when a CUDA device is available.

At inference time, each sentence is tokenized with truncation and padding, and the most likely label is selected for each token. Term candidates are decoded by grouping consecutive B-TERM/I-TERM tokens into spans, which are converted back to surface strings via tokenizer detokenization. Finally, the extracted strings are cleaned by trimming the surrounding punctuation, normalizing whitespace, and lowercase, to comply with the task constraints.

2.3. spaCy sequence labeller

As a complementary extractor, we train a spaCy pipeline [10] by defining term extraction as an NER task with a TERM of a single entity type. We start from the Italian pretrained model `it_core_news_md` when available, and train (update) the ner component only, while disabling the other pipeline components during optimization.

Training annotations are derived from the gold term lists by searching for each term as a substring of the sentence and converting matched character spans into spaCy entities through `Doc.char_span`, using `alignment_mode=expand` to ensure valid token boundaries. Since multiple matches may overlap, we apply an overlap-resolution heuristic that keeps longer spans when conflicts occur. To focus learning on informative examples, we discard sentences that contain no annotated TERM spans.

We train on the union of the official training and development sets for 40 iterations with dropout 0.1 and batch size 8. At inference time, we collect TERM spans from `doc.ents`, lowercase them, and remove duplicates at sentence level before the ensemble stage (Section 2.4).

2.4. Vocabulary-based filtering and merging

The final system combines the raw term candidates produced by the BERT extractor (Section 2.2) and the spaCy sequence labeler (Section 2.3) through a lightweight ensemble guided by a gold-derived domain vocabulary. The predictions of the two models are aligned at sentence level using the shared identifiers (`document_id`, `paragraph_id`, `sentence_id`).

All terms are first canonicalized with a shared normalization function that converts the text to lowercase, applies Unicode normalization (NFKC), unifies apostrophes/quotes, collapses whitespace, and strips punctuation at the term boundaries. This normalization is applied to both predictions and gold terms so that matching and filtering are performed consistently.

Gold-derived vocabulary. We build a vocabulary of domain terms from the official training and development annotations (train + dev) by collecting all gold terms and storing their normalized forms. The resulting set acts as a domain whitelist used to validate candidates and to decide which spaCy spans are reliable, reducing the inclusion of well-formed but out-of-domain expressions.

Multiword upgrade (BERT → spaCy). BERT occasionally predicts short fragments that are part of a longer multiword expression (e.g., predicting “ferro” instead of “materiali ferrosi”). To address this, we attempt to “upgrade” each BERT term to a longer spaCy span when this is supported by the gold vocabulary. Concretely, for each BERT prediction, we search among spaCy candidates for the longest multiword span that (i) is present in the gold-derived vocabulary and (ii) contains the BERT term as a contiguous token subspan; if found, the BERT term is replaced by the longer span. We also maintain a small blacklist of clearly non-terminological items (e.g., time expressions and generic discourse words) that are discarded before and after upgrading.

Dictionary-filtered addition of spaCy spans. After upgrading, we enrich the output by adding additional spaCy multiword candidates that are present in the gold-derived vocabulary, are not already selected, and are not blacklisted. This step primarily increases recall for well-formed multiword terms that spaCy detects reliably but that may be missed by BERT, while keeping precision under control via the whitelist.

Generic unigram filtering and final formatting. Finally, we filter out overly generic single-word candidates using a small set of generic heads (e.g., *servizio*, *materiali*, *impianto*) when they do not appear in the gold vocabulary. Acronym-like strings (short alphabetic forms, optionally with dots) are preserved. The remaining terms are normalized and de-duplicated at sentence level using their canonical form, preserving the original output order (upgraded BERT terms first, then added spaCy spans). The final list is emitted in the required ATE-IT JSON format.

2.5. Post-processing

After merging, we apply a deterministic post-processing stage to enforce ATE-IT output constraints and remove systematic noise. Given a sentence and its raw predicted term list, we perform the following steps:

- **String normalization and sentence-level de-duplication** Each candidate term is normalized by trailing whitespace, collapsing multiple spaces, stripping punctuation only at the boundaries, and lowercasing. Empty strings are discarded. We then remove duplicates within the same sentence while preserving the original order.
- **Truncated-term filtering** Both extractors occasionally output incomplete multiword terms that end with a function word (e.g., “*gestione dei*”, “*batterie e*”), typically caused by boundary errors in span decoding. We filter these cases with a simple heuristic: a candidate is marked as truncated if it has at least two tokens, and its last token belongs to a predefined set of Italian function-word endings (prepositions and conjunctions such as “*di*”, “*dei*”, “*e*”, “*a*”, etc.). All candidates flagged as truncated are removed.
- **Nested-term removal using character spans** The task explicitly excludes nested terms within the same sentence [4]. To enforce this constraint, we remove candidates that are strictly contained inside longer candidates in the sentence texts. For each candidate, we locate all (case-insensitive) occurrences in the sentence and represent them as spans (*start*, *end*). We then sort all spans by decreasing length and apply a greedy selection: a candidate span is kept only if it is not fully contained in any already selected span. Finally, selected spans are ordered by their start offset to preserve sentence order, and we return the corresponding unique terms. This step avoids outputting both a long term and its inner subspans when they refer to the same textual occurrence.
- **Output format** After cleaning, the per-sentence term list is written in the official ATE-IT JSON format: {*document_id*, *paragraph_id*, *sentence_id*, *term_list*}. This guarantees lowercase-only terms, sentence-level de-duplication, and compliance with the no-nesting constraint.
- **Final submission sanity-check: token-based nesting removal** Before producing the final submission file, we run an additional lightweight nested-term check directly on each *term_list*, without using the original sentence text. A term is considered nested if its token sequence appears as a contiguous subsequence of a longer term in the same list (e.g., *trattamento rifiuti* inside *impianto di trattamento rifiuti*). All nested (shorter) terms are removed while maintaining order. The script also reports an approximate count of nested pairs before and after this pass, acting as a final safeguard against residual nesting errors.

3. Results

Data splits. During development, we trained models on the official training set and evaluated them on the official development set (`subtask_a_dev.json`) to select hyperparameters and pipeline variants. After selecting the final approach, we retrained the supervised components in a *train+dev* setting and ran inference on the test set to generate the official submission.

Evaluation protocol. We evaluate systems using the ATE-IT official metrics [4] by comparing predicted and gold term lists at sentence level using: (i) **Micro** Precision/Recall/F1 (aggregating true positives, false positives, and false negatives across all sentences) [5] and (ii) **Type** Precision/Recall/F1 (computed over the set of unique predicted vs. unique gold term types, ignoring frequency) .

Systems compared. We compared four families of approaches:

- **LLM baselines** (zero-shot / few-shot): `llm_zero_shot`, `llm_few_shot`.
- **Rule-based baselines:** `vanilla`, and a `spaCy` pattern-based extractor (`spacy_term_extraction_patterns`).
- **Supervised extractors:** fine-tuned BERT token classifier (Section 2.2) and a `spaCy` NER tagger trained on ATE-IT data (Section 2.3).
- **Ensembles:** BERT + `spaCy` merging with gold-vocabulary filtering (Section 2.4), also followed by LLM reranking.

3.1. LLM baselines and reranking: models and prompts

We report the exact models and prompt templates used for the prompt-based baselines and for LLM reranking.

llm_zero_shot. Model: `gemini-2.5-flash` via `google.generativeai`. We process sentences in batches of 20 and prompt the model to output one list of terms per sentence in a fixed textual format. The system prompt is:

You are an automatic term extraction agent. You will receive a list of sentences as input. Your role is to extract waste management terms from the sentences. Output a list of terms for each sentence.

Strictly adhere to the Example Output Format:

Example Output Format: Sentence 1: [term1; term2; term3; term4] Sentence 2: [term5; term6]
Sentence 3: [] Sentence 4: [] Sentence 5: [term7]

Instructions: - Extract only terms, ignore named entities - Do not extract nested terms - Extract only terms related to waste management, ignoring other domains - If a sentence contains no terms, output an empty list for that sentence - You must output 20 lists of terms, one for each sentence

llm_few_shot. Model: `gemini-2.5-flash` via `google.generativeai`. We use the same batching and output format as in zero-shot, but we prepend annotated examples (few-shot) to steer extraction and constraint compliance. The system prompt is:

You are an automatic term extraction agent for Italian municipal waste management texts. You will receive a list of sentences as input. Your role is to extract **all and only** waste management terms from each sentence. Output a list of terms for each sentence.

A "term" in this task is: - a single- or multi-word expression - refers to a concept in the waste management domain - often nouns or noun phrases (sometimes adjectives or verbs as part of a phrase)

Non-terms are: - generic function words (e.g., "e", "di", "per", "che") - pure numbers or dates not part of a waste term - person names, city names, street names (unless part of an official name of a waste service)

Strictly adhere to the Example Output Format:

Sentence 1: [term1; term2; term3] Sentence 2: [term4] Sentence 3: []

----- ANNOTATED EXAMPLES (FEW-SHOT) -----

Example 1 Sentence: "Il presente disciplinare per la gestione dei centri di raccolta comunali è stato redatto ai sensi del DM 13/05/2009." Expected output: Sentence 1: [disciplinare per la gestione dei centri di raccolta comunali; centri di raccolta comunali]

Example 2 Sentence: "Il servizio di raccolta differenziata porta a porta dei rifiuti urbani è attivo su tutto il territorio comunale." Expected output: Sentence 1: [servizio di raccolta differenziata porta a porta; raccolta differenziata porta a porta; rifiuti urbani]

Example 3 Sentence: "Il pagamento della Tassa Rifiuti (TARI) avviene tramite il portale pagoPA." Expected output: Sentence 1: [tassa rifiuti; tari]

Example 4 Sentence: "Il presente regolamento disciplina le modalità di conferimento dei rifiuti ingombranti presso l'isola ecologica comunale." Expected output: Sentence 1: [regolamento; modalità di conferimento; rifiuti ingombranti; isola ecologica comunale]

Example 5 Sentence: "In questa frase non sono presenti termini di gestione dei rifiuti." Expected output: Sentence 1: []

YOUR TASK: Now you will receive 20 sentences in the following format:

Sentence k: <sentence_text>

For each sentence k, you MUST output exactly one line in this format:

Sentence k: [term1; term2; term3]

Instructions: - Extract only terms related to waste and waste management (e.g., *tassa rifiuti*, *tari*, *isola ecologica*, *raccolta differenziata*, *impianto di trattamento rifiuti*). - Prefer complete multi-word terms (full span) over shorter fragments. - Do NOT output nested terms: if you extract "*impianto di trattamento rifiuti urbani*", do NOT also output "*trattamento rifiuti urbani*", unless it appears as a separate term in the sentence. - Ignore named entities (people, cities, streets) unless they are part of an official waste management term. - If a sentence contains no relevant terms, output an empty list: Sentence k: []. - You must output one line for each of the 20 input sentences, in order (Sentence 1, Sentence 2, ...).

LLM reranking. Models (Groq-hosted): `openai/gpt-oss-120b`, `openai/gpt-oss-20b`, `qwen/qwen3-32b`. Given a sentence and a candidate term list (from supervised extractors), the reranker assigns a score in $[0, 1]$ and a keep/discard decision for each candidate, returning strict JSON. The system prompt is:

You are an automatic term extraction reranking agent for Italian municipal waste management texts.

You will receive: - one sentence in Italian, and - a list of candidate terms extracted by a baseline system.

Your task: - For each candidate term, decide if it is a good domain-relevant term in the context of the sentence. - Assign a relevance score between 0.0 and 1.0 (higher = better). - Decide whether to keep or discard each candidate.

----- DOMAIN KNOWLEDGE (WASTE MANAGEMENT) -----

Valid waste management terms typically include:

- 1) Waste types and materials - *rifiuti*, *rifiuti urbani*, *rifiuti ingombranti*, *rifiuti pericolosi*, *rifiuti organici* - *carta*, *cartone*, *plastica*, *vetro*, *metalli*, *alluminio*, *banda stagnata*, *legno*, *tessili* - *RAEE* / *R.A.E.E.*, *toner*, *oli esausti*
- 2) Services, infrastructures, processes - *raccolta differenziata*, *raccolta porta a porta*, *modalità di raccolta*, *modalità di conferimento* - *servizio di raccolta*, *servizio di igiene urbana* - *centro di raccolta*, *centri di raccolta comunali*, *isola ecologica*, *ecocentro*, *piattaforma ecologica* - *impianto di trattamento rifiuti*, *impianto di smaltimento*, *discarica*, *compostaggio*

System (dev)	P_μ	R_μ	$F1_\mu$	P_t	R_t	$F1_t$
<i>Ensembles (BERT + spaCy + vocabulary)</i>						
ensemble_bert_2e-5_spacy_dictfilter	0.770	0.734	0.751	0.728	0.674	0.700
ensemble_bert_2e-5_spacy_baseline_dictfilter	0.755	0.736	0.745	0.703	0.686	0.695
ensemble_bert_spacy (no vocab filter)	0.640	0.703	0.670	0.545	0.628	0.583
<i>LLM reranking (on top of supervised candidates)</i>						
ensemble_bert_2e-5_reranked_llm	0.782	0.701	0.739	0.760	0.653	0.702
ensemble_bert_2e-5_cased_reranked_llm	0.756	0.714	0.734	0.695	0.669	0.682
<i>BERT-only tuning</i>						
bert_2e-5_cleaned	0.758	0.710	0.733	0.714	0.649	0.680
bert_2e-5	0.732	0.710	0.721	0.677	0.649	0.662
bert_3e-5_cleaned	0.734	0.681	0.707	0.684	0.607	0.643
<i>Baselines and spaCy-only</i>						
llm_few_shot	0.463	0.665	0.546	0.404	0.719	0.517
llm_zero_shot	0.481	0.630	0.545	0.408	0.653	0.502
vanilla	0.428	0.738	0.542	0.657	0.562	0.606
spacy_trained	0.591	0.353	0.442	0.554	0.380	0.451
spacy_baseline	0.502	0.350	0.413	0.701	0.368	0.482
spacy_term_extraction_patterns	0.051	0.373	0.090	0.041	0.417	0.074

Table 1

Development-set results for the main exploratory runs. Micro metrics (subscript μ) aggregate matches across sentences; Type metrics (subscript t) are computed over unique term types [4].

3) Regulations, tariffs, administrative terms - regolamento, disciplinare, disciplinare per la gestione dei centri di raccolta comunali - tassa rifiuti, TARI, tariffa puntuale, tariffe - utenti, utenze domestiche, utenze non domestiche

4) Action verbs and phrases that are domain terms - conferire, conferiti, vanno conferiti, conferimento, modalità di conferimento - depositare, deposito, conferire i rifiuti, conferire i rifiuti vegetali

Non-terms are: - generic function words (es. "e", "di", "per", "che") - pure numbers or dates not part of a waste term - person names, city names, street names (unless part of an official waste management service) - clearly truncated fragments (e.g., "dei", "oo", "r" alone)

----- OUTPUT FORMAT (STRICT JSON) -----

You MUST output ONLY a JSON object with this exact structure:

"reranked_terms": ["term": "...", "score": 0.0, "keep": true or false, ...]

Rules: - Do not add new terms that are not in the candidate list. - Do not modify the spelling of the candidates. - If a candidate looks truncated or not a full concept, set "keep": false and give it a low score. - If no candidate is good, you may return an empty list: "reranked_terms": []. - The JSON must be valid and parseable by Python's json.loads().

Your scores will be later combined with a rule-based domain scorer, so: - assign higher scores to candidates that clearly match the domain knowledge above, - and lower scores to generic or borderline expressions.

The corresponding user prompt provides the sentence and the candidate list (one per line) and asks the model to output only the JSON object described above.

3.2. Exploratory results on development set

Table 1 reports the main exploratory runs on the development set. For readability, we group results by model family and highlight the best configurations.

Exploratory configurations. Before selecting the final SMTE pipeline, we varied both the supervised extractors and the ensemble strategy. For BERT, we compared *uncased* vs. *cased* Italian checkpoints and

multiple fine-tuning settings (e.g., learning rate $2 \cdot 10^{-5}$ vs. $3 \cdot 10^{-5}$), and we measured the effect of deterministic cleaning on span decoding. For spaCy, we trained the TERM tagger starting from different Italian backbones (`it_core_news_sm` and `it_core_news_md`) and compared a trained tagger against a lighter baseline component. Finally, we tested multiple integration variants: (i) dictionary-guided merge (BERT+spaCy+vocabulary), (ii) a weaker merge using spaCy baseline spans, and (iii) LLM reranking on top of supervised candidates.

Discussion of the results. Results show a clear separation between *supervised* approaches trained on ATE-IT annotations and *prompt-based* / purely rule-based baselines. The best exploratory system is the supervised ensemble `ensemble_bert_2e-5_spacy_dictfilter` (Micro-F1 = 0.751; $P_\mu = 0.770$, $R_\mu = 0.734$), improving over the strongest single-component model, BERT-only with deterministic cleaning (`bert_2e-5_cleaned`, Micro-F1 = 0.733). This gain comes from combining three complementary signals: (i) BERT provides accurate contextual supervision for term boundaries, (ii) spaCy contributes additional syntactically well-formed multiword candidates, and (iii) the gold-derived vocabulary acts as a domain constraint that prevents spaCy expansions from introducing many out-of-domain false positives. A key mechanism is *multiword upgrading*: when BERT predicts a short fragment, it can be replaced with a longer spaCy span only if the resulting multiword expression is present in the gold-derived vocabulary, improving span completeness while keeping precision controlled.

Post-processing is consistently beneficial: for the same BERT configuration, Micro-F1 increases from 0.721 (`bert_2e-5`) to 0.733 (`bert_2e-5_cleaned`), mainly by removing truncated spans, duplicates, and nested outputs. Moreover, the vocabulary constraint is crucial: simply pooling the BERT+spaCy candidates without filtering (`ensemble_bert_spacy`, Micro-F1 = 0.670) is substantially worse than dictionary-guided merge. Finally, LLM baselines remain around Micro-F1 ≈ 0.545 and show the typical high-recall/low-precision pattern, suggesting that without supervised adaptation they struggle to match span guidelines and domain constraints, so they were later discarded.

LLM reranking does not outperform the vocabulary-filtered ensemble: for example, the `ensemble_bert_2e-5_reranked_llm` produces Micro-F1 = 0.739, increasing precision ($P_\mu = 0.782$) but decreasing recall ($R_\mu = 0.701$), the reranker acts mainly as a conservative filter that removes both false positives and some true terms, so the loss in recall outweighs the precision gain, and the overall F1 remains below the best dictionary-guided merge.

3.3. Final re-run and submission-ready outputs

After selecting the best approach (BERT + spaCy + vocabulary merge) and the best models and hyperparameters, a final re-run was performed using simplified names and explicitly checked compliance with ATE-IT output constraints (in particular, *no nested terms*). Table 2 reports this final re-run:

- **final_MD** / **final_SM** are the ensemble outputs after the BERT+spaCy+vocabulary merge (Section 2.4), where spaCy is trained starting from `it_core_news_md` vs. `it_core_news_sm`.
- **final_MD_submission** / **final_SM_submission** are the corresponding outputs after applying `clean_final_submission.ipynb`, which performs an additional token-based nested-term removal on each sentence `term_list` as a final safeguard.

Note on scores (model selection vs. submission-ready output). The best exploratory configuration (`ensemble_bert_2e-5_spacy_dictfilter`) reaches Micro-F1 = 0.751 on the development set. However, after manual inspection, we observed residual violations of the official ATE-IT output constraints, in particular the presence of nested terms within the same sentence-level `term_list`. Our original post-processing already performs string normalization, sentence-level de-duplication, truncated-span filtering, and nested-term removal by locating term occurrences in the sentence text (character-span based filtering). Nevertheless, a small number of token-level nesting cases can survive because span matching is sensitive to edge cases, such as punctuation/whitespace variants, partial

System (dev)	P_μ	R_μ	$F1_\mu$	P_t	R_t	$F1_t$
final_MD	0.765	0.727	0.745	0.718	0.674	0.695
final_SM	0.743	0.725	0.734	0.688	0.674	0.681
final_MD_submission	0.773	0.723	0.747	0.725	0.665	0.694
final_SM_submission	0.774	0.721	0.746	0.725	0.665	0.694
bert_2e-5	0.732	0.710	0.721	0.677	0.649	0.662
spacy_trained_MD	0.598	0.339	0.433	0.585	0.384	0.464
spacy_trained_SM	0.476	0.326	0.387	0.405	0.368	0.385
spacy_baseline_MD	0.289	0.341	0.313	0.349	0.360	0.354
spacy_baseline_SM	0.289	0.341	0.313	0.349	0.360	0.354

Table 2

Final re-run on development set: ensemble outputs and submission-cleaned outputs. The *_submission variants include a final token-based nesting-removal safeguard.

System	P_μ	R_μ	$F1_\mu$	P_t	R_t	$F1_t$
Baseline	0.497	0.559	0.526	0.435	0.508	0.469
SMTE (ours)	0.656	0.577	0.614	0.645	0.529	0.581

Table 3

Official ATE-IT Subtask A results on the hidden test set for our submitted system (SMTE) and the shared-task baseline [11]. The table reports both Micro-level (P_μ , R_μ , $F1_\mu$) and Type-level (P_t , R_t , $F1_t$) evaluation metrics.

matches, or multiple occurrences of the same term in the sentence. For the final submission artifact, we therefore added a deterministic “safety” pass that removes nested terms directly at the `term_list` level via token-subspan containment checks (e.g., dropping *trattamento rifiuti* when *impianto di trattamento rifiuti* is also present in the list). This produces a submission-ready output that strictly enforces the constraints and slightly changes the precision/recall balance, yielding Micro-F1 = 0.747 on dev set. We report both values to clearly distinguish *model selection* from the final *constraint-compliant submission output*.

3.4. Official results on the test set

After selecting the final configuration, we trained the supervised components on the union of the official training and development data (*train+dev*) and generated predictions for the official hidden test set, which was evaluated by the shared-task organizers. Table 3 reports the official leaderboard scores for our submitted run (SMTE), together with the baseline provided for reference [11].

Development-set scores (Section 3.2) are used for model selection and ablations, while the official results are computed on a hidden test set. The drop from dev to test is expected and reflects generalization to unseen data (e.g., different term distribution and sentence formulations).

4. Discussion

This section summarizes what worked well in SMTE, what did not, and which improvements are most promising for future iterations.

4.1. What worked

Supervised extraction is essential. Development experiments show that supervised sequence labeling is the key factor for competitive performance in ATE-IT. Both BERT and the spaCy tagger benefit from direct training on the task annotations, which encode not only domain terminology, but also the shared-task span conventions. In contrast, prompt-based LLM baselines and purely rule-based systems tend to over-generate plausible candidates and struggle to consistently match the gold span boundaries and selection rules.

The ensemble improves span completeness and robustness. The best results are obtained by combining complementary signals: BERT contributes high-quality contextual boundaries, while spaCy contributes syntactically coherent multiword candidates. The ensemble logic is especially effective when BERT predicts an incomplete fragment of a multiword expression: the *multiword upgrade* step replaces short predictions with longer spaCy spans when supported by the gold-derived vocabulary. This mechanism improves term completeness without inflating false positives, yielding a better precision/recall balance than either component alone.

Deterministic cleaning and constraint enforcement reduce false positives. Post-processing steps (normalization, duplicate removal, truncated-span filtering, and nested-term removal) consistently improve quality. In this task, compliance checks are not only formatting details: violations, such as truncated terms or nested output, directly translate into false positives under the official evaluation. The final submission-cleaning step provides an additional safeguard by removing residual nested terms at the list level, ensuring that the output respects the ATE-IT constraints in a deterministic and reproducible way.

4.2. What did not work

LLM reranking is not reliably beneficial. LLM reranking slightly increases precision in some runs by discarding questionable candidates, but it does not improve Micro-F1 over the best vocabulary-filtered merge. The most common failure mode is a recall drop: reranking can remove correct terms when the model is uncertain about guideline-specific decisions (e.g., whether a generic-looking span is acceptable in this domain).

spaCy-only systems have limited recall and boundary noise. The spaCy trained tagger improves over baseline spaCy components, but alone it remains substantially weaker than BERT-based extraction. Typical issues are missed domain terms (especially less frequent expressions) and boundary mismatches (e.g., including or excluding function words at the edges of a term). This confirms the utility of spaCy primarily as a *complementary candidate generator* within the ensemble, rather than as a standalone extractor.

4.3. Error analysis and limitations

Boundary errors and partial spans. Despite cleaning, span boundary errors remain a core challenge: models may include trailing function words (“gestione dei”), misleading modifiers, or output truncated multiword expressions. These errors are amplified by sentence-level evaluation, where small boundary differences make a prediction count as incorrect.

Generic terms vs. domain-specific usage. The municipal waste-management domain contains many words that are meaningful in context but generic in isolation (e.g., servizio, gestore, materiali). Even with generic-header filtering and a gold-derived vocabulary, distinguishing true terminology from generic discourse remains difficult, especially for single-word candidates. This affects both precision (generic false positives) and recall (over-filtering legitimate generic-looking terms).

Nested terms and overlapping candidates. The no-nesting constraint is strict and sometimes ambiguous in practice: a shorter span can be a valid term type but becomes invalid if it is nested inside a longer term in the same sentence occurrence. This requires sentence-aware reasoning and careful implementation.

Vocabulary dependence and domain drift. The vocabulary filter substantially improves robustness, but it introduces a limitation: it is built from train and development gold terms, so it may under-represent

rare or unseen terminology in the test set. This can cap recall when new domain expressions appear (domain drift) or when gold annotations include valid variants not present in the training vocabulary.

4.4. Future work

Several extensions could further improve SMTE system:

- **Constraint-aware decoding.** Integrate “no nesting” directly into decoding (or candidate selection) instead of enforcing it after generation, e.g., by selecting a globally consistent set of non-overlapping spans per sentence.
- **Stronger span modelling.** Replace token-level BIO with span-based methods (e.g. pointer/span classification) to reduce boundary mismatches and truncated output.
- **Richer domain resources;** Expand the vocabulary with external terminological sources (e.g., waste-management glossaries) and add controlled normalization rules for abbreviations and orthographic variants.
- **Better calibration for generic terms;** Replace hard-coded generic unigram filtering with a learned classifier (or calibrated scoring) that uses contextual features to decide when generic-looking words are terminological.
- **Using more context;** The task is sentence-based, but lightweight document context (e.g., previous sentence, section title when available) could help disambiguate generic candidates and improve consistency.

In summary, the final system demonstrate that the most reliable strategy for ATE-IT is a supervised ensemble with deterministic constraint enforcement: BERT provides strong contextual supervision, spaCy adds multiword robustness, and vocabulary-guided filtering plus post-processing ensures both higher precision and compliance with the task output requirements.

Declaration on Generative AI

In this work Generative AI tools were used during development to assist with coding (drafting small code snippets, refactoring suggestions, and documentation wording) and to support the writing of parts of the report (e.g., wording refinements and clarity improvements). All suggested code was manually reviewed, debugged, and adapted to the project constraints, and the final submission consists of runnable notebooks verified by the author.

References

- [1] G. M. Di Nunzio, S. Marchesin, G. Silvello, A systematic review of Automatic Term Extraction: What happened in 2022?, *Digital Scholarship in the Humanities* 38 (2023) i41–i47. doi:10.1093/dsh/11c/fqad030.
- [2] B. Jehangir, S. Radhakrishnan, R. Agarwal, A survey on Named Entity Recognition — datasets, tools, and methodologies, *Natural Language Processing Journal* 3 (2023) 100017. doi:10.1016/j.nlp.2023.100017.
- [3] F. Cutugno, A. Miaschi, A. P. Apro시오, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [4] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.

- [5] R. Verborgh, M. Röder, R. Usbeck, A.-C. Ngonga Ngomo, Gerbil – benchmarking named entity recognition and linking consistently, *Semantic Web 9* (2018) 605–625. doi:10.3233/SW-170286.
- [6] M. Martinelli, Ata_tutorship: Tutorship baseline code and reference implementations for ate-it, GitHub repository, 2025. URL: https://github.com/MMartinelli-hub/ATA_Tutorship.
- [7] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: *Proceedings of NAACL-HLT, 2019*, pp. 4171–4186. doi:10.18653/v1/N19-1423.
- [8] L. Ramshaw, M. Marcus, Text Chunking using Transformation-Based Learning, in: *Third Workshop on Very Large Corpora*, 1995.
- [9] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, Transformers: State-of-the-art natural language processing, in: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020*, pp. 38–45. doi:10.18653/v1/2020.emnlp-demos.6.
- [10] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spacy: Industrial-strength natural language processing in python, Website, 2025. URL: <https://spacy.io/>.
- [11] ATE-IT Shared Task, Ate-it results (evalita 2026) — official leaderboard, Website, 2025. URL: <https://nicolacirillo.github.io/ate-it/results.html>.

TermNinjas at ATE-IT: Overview of the Term Extraction and Term Variants Clustering Task

Zunaira Hasnain¹, Fashad Ahmed Siddque¹

¹*Department of Computer Science and Engineering (DISI), Artificial Intelligence Programme, University of Bologna, Italy*

Abstract

This paper presents a hybrid neural symbolic system submitted to the ATE-IT Shared Task at EVALITA 2026, addressing both Subtask A (Automatic Term Extraction) and Subtask B (Term Variants Clustering) on Italian municipal waste management documents. The proposed approach combines supervised fine tuning of an Italian BERT model for sequence labeling with extensive domain aware post processing, followed by a multi stage clustering pipeline for grouping term variants.

For Subtask A, the system focuses on high precision extraction of single word and multi word domain terms through BIO tagging and constraint aware reconstruction. For Subtask B, a hybrid clustering strategy is adopted, integrating lemmatization and multilingual semantic embeddings, with optional large language model based verification. Experimental results on the official test set show competitive performance, with particularly strong precision and clear improvements over the official baseline in the clustering task. The entire system relies exclusively on open source tools and models, ensuring full reproducibility without external API dependencies.

Keywords

Automatic Term Extraction, Term Variants Clustering, Italian NLP, Waste Management, BERT, Hybrid Systems

1. Introduction

Automatic Term Extraction (ATE) is a fundamental task in Natural Language Processing (NLP) aimed at identifying domain specific terminology from specialized corpora. Extracted terms represent the conceptual backbone of a domain and are widely used in downstream applications such as ontology construction, knowledge base population, terminology management, domain specific information retrieval, and technical document indexing. Unlike Named Entity Recognition (NER), which focuses on proper names and referential entities, ATE targets common nouns and multi word expressions that encode domain knowledge rather than unique identifiers.

The ATE-IT Shared Task at EVALITA 2026 [1] introduces the first large scale evaluation benchmark for Automatic Term Extraction and Term Variants Clustering in Italian. The task focuses on institutional documents related to municipal waste management, a domain characterized by formal administrative language, complex noun phrases, frequent use of acronyms, and extensive lexical variation. These properties make the task particularly challenging for purely statistical or end to end neural approaches.

The shared task is composed of two subtasks. Subtask A addresses Automatic Term Extraction at the sentence level, requiring systems to identify both single word and multi word domain terms under strict constraints. In particular, nested terms are forbidden, surface forms must match exactly, and duplicate extractions within the same sentence are penalized. Subtask B focuses on Term Variants Clustering, where systems must group extracted terms that refer to the same underlying concept, while explicitly avoiding the clustering of hypernym hyponym relations.

In this paper, we present the unified hybrid neural symbolic framework designed to address both subtasks. Our approach combines supervised transformer based sequence labeling with domain aware symbolic post processing for Subtask A, followed by a multi stage clustering pipeline integrating linguistic normalization, semantic embeddings, and strict anti lumping controls for Subtask B. A key design goal of the system is full reproducibility: all components rely exclusively on open source models and libraries and can be executed locally without external API dependencies.



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Related Work

Early approaches to Automatic Term Extraction relied primarily on statistical association measures such as TF IDF, mutual information, log likelihood ratios, and C-value. These methods often combined frequency based scoring with shallow linguistic filters, such as part-of-speech patterns, to identify candidate terms. While effective in controlled settings, these approaches struggle with contextual ambiguity, long multi word expressions, and domain specific paraphrases.

More recent work formulates ATE as a sequence labeling problem, enabling the use of neural architectures originally developed for Named Entity Recognition. Recurrent neural networks and, more recently, transformer based models such as BERT have demonstrated strong performance due to their ability to capture contextual information. However, purely neural approaches often suffer from overgeneration, producing noisy or partially correct spans that violate task specific constraints such as non nesting and sentence level uniqueness.

Hybrid approaches combining neural models with symbolic post processing have been proposed to address these limitations. By enforcing linguistic and structural constraints after neural prediction, such systems improve precision and interpretability without sacrificing the benefits of contextualized representations.

Term Variants Clustering has traditionally combined linguistic normalization techniques, such as lemmatization and stemming, with vector space similarity measures. With the advent of sentence and phrase embeddings, semantic similarity has become a dominant signal for clustering. Evaluation is commonly performed using BCubed Precision, Recall, and F1, which assess both cluster purity and completeness. Recent baselines explore zero shot large language models for clustering, but these approaches raise concerns regarding reproducibility, computational cost, and controllability.

3. System Description

The system is designed as a modular pipeline addressing both subtasks of the ATE-IT Shared Task. The system follows a hybrid neural symbolic paradigm, where neural models are responsible for capturing contextual and semantic information, while symbolic components enforce task specific constraints and domain knowledge.

For Subtask A, Automatic Term Extraction is modeled as a BIO sequence labeling problem using a fine tuned Italian BERT model. Neural predictions are refined through constraint aware decoding and extensive post processing to enforce non nesting, sentence level uniqueness, and domain relevance.

For Subtask B, extracted terms are clustered using a hybrid approach that combines lemmatization based grouping with semantic similarity computed from multilingual sentence embeddings. A strict clustering configuration introduces additional rule based modules to prevent overly broad or semantically incoherent clusters.

The overall pipeline is fully reproducible and relies exclusively on open source tools, including HuggingFace Transformers, SpaCy, and sentence transformers. No proprietary APIs or external services are required.

3.1. Design Principles

This system is designed around three core principles: (i) precision oriented extraction suitable for downstream terminology management tasks, (ii) modularity, enabling independent experimentation across subtasks, and (iii) full reproducibility using exclusively open source components. Rather than relying on end-to-end generative approaches, the system explicitly separates linguistic modeling from symbolic constraints, allowing finer control over error sources and interpretability.

3.2. Overall System Architecture

This system follows a two stage hybrid neural symbolic pipeline. In the first stage, raw text is processed by a supervised transformer based sequence labeling model to identify candidate term spans. These predictions are subsequently refined through constraint aware decoding and domain specific symbolic filtering to ensure task compliance and terminological precision.

In the second stage, the set of unique extracted terms is passed to a modular clustering pipeline. Linguistic normalization and lemmatization are first applied to capture inflectional variants. Semantic similarity is then computed using multilingual sentence embeddings, and clusters are refined through threshold based merging and explicit anti lumping mechanisms. Optional large language model verification can be applied to ambiguous cases, though it is disabled in the official submission to preserve efficiency and reproducibility.

3.3. Subtask A: Automatic Term Extraction

Subtask A addresses Automatic Term Extraction (ATE) from Italian institutional documents related to municipal waste management. The objective is to identify domain specific terminology, including both single word terms and multi word expressions, under strict task constraints prohibiting nested terms and duplicate extractions within the same sentence. Unlike Named Entity Recognition, the target expressions do not follow canonical entity patterns and frequently involve compositional noun phrases, technical jargon, and administrative formulations.

3.3.1. Data Analysis and Task Characteristics

The training data exhibits substantial sparsity and imbalance. Only a subset of sentences contains annotated terms, and term tokens are heavily outnumbered by non term tokens. The average term length exceeds two tokens, with a long tail of complex multi word expressions reaching up to twenty tokens. These properties motivate a sequence labeling formulation combined with explicit constraint enforcement to control overgeneration.

3.3.2. Preprocessing and Linguistic Normalization

All sentences are lowercased to ensure case insensitive matching, as required by the evaluation protocol. Bracketed content is preserved while removing enclosing characters, and excessive whitespace is normalized. Tokenization is performed using the `it_core_news_sm` SpaCy model, which provides robust handling of Italian morphology, contractions, and punctuation. No lemmatization or stemming is applied at this stage to avoid altering surface forms required for exact match evaluation.

3.3.3. BIO Encoding and Gold Alignment

Gold standard terms are converted into a BIO tagging scheme with labels B-TERM, I-TERM, and O. To correctly handle overlapping and nested annotations in the training data, terms are processed using a longest first strategy: longer multi word expressions are encoded before shorter ones, preventing fragmentation of maximal spans. Token level BIO labels are aligned with transformer subword tokenization using word index mappings, where only the first subword of each token contributes to the loss and subsequent subwords are masked.

3.3.4. Neural Model and Training Configuration

The extraction model is based on `dbmdz/bert-base-italian-uncased`, fine tuned for token classification. The model is trained for five epochs with a batch size of 16 and a learning rate of 2×10^{-5} using the Adam optimizer. Training is performed in a fully supervised manner without any prompting, generative inference, or external language model interaction. Random seeds are fixed to ensure reproducibility.

Due to the extreme dominance of the O label, token level predictions tend to be conservative. While class weighted loss is supported, the primary strategy for mitigating this imbalance is applied at inference time through probability aware decoding rather than hard argmax selection. This design choice allows borderline B-TERM predictions to be retained, improving recall without retraining the model.

3.3.5. Inference and Probability Based Decoding

At prediction time, token level logits are converted into label assignments with careful alignment back to SpaCy tokens. Only one label per original token is retained. Instead of relying exclusively on the most probable label, probability thresholds are explored to capture low confidence but semantically plausible term boundaries. This approach is particularly beneficial for long multi word expressions, where uncertainty accumulates across tokens.

3.3.6. Constraint Aware Term Reconstruction

Predicted BIO sequences are reconstructed into surface form terms by merging contiguous B-TERM and I-TERM spans. A multi stage post processing pipeline is then applied to enforce ATE-IT constraints and remove systematic noise. Duplicate terms within the same sentence are eliminated. Nested terms are removed unless the shorter term occurs independently outside the span of a longer term in the same sentence. This check is implemented using span level matching over the original sentence text.

3.3.7. Domain Specific Filtering

To further improve precision, extracted candidates are filtered using a set of domain aware rules. These include the removal of Italian stopwords, generic verbs, administrative headers, days of the week, isolated prepositions, incomplete fragments, and non domain English terms. Acronyms are retained only if they belong to a curated list of waste management related abbreviations (e.g., RAEE, TARI). Formatting normalization corrects spacing around punctuation and Italian contractions, ensuring consistent surface forms.

3.3.8. Evaluation Protocol

Evaluation follows the official ATE-IT specification. Micro F1 is computed at the sentence level by comparing sets of extracted terms against gold annotations, aggregating true positives, false positives, and false negatives across the corpus. Type F1 evaluates unique term types globally, measuring the system's ability to recover the domain vocabulary independently of sentence context. All comparisons are performed in a case-insensitive manner without lemmatization.

3.4. Subtask B: Term Variants Clustering

Subtask B focuses on clustering term variants that refer to the same underlying concept within the Italian municipal waste management domain. Given a set of unique terms extracted in Subtask A, the goal is to group together inflectional variants, acronyms and their expansions, and true synonyms, while explicitly avoiding the clustering of hypernym-hyponym relations (e.g., *rifiuti* vs. *rifiuti indifferenziati*). The task is evaluated using BCubed Precision, Recall, and F1, which measure cluster coherence and completeness at the item level.

3.4.1. Overview of the Clustering Pipeline

We implement a modular and configurable production pipeline that supports multiple clustering strategies, including lemma based, embedding based, LLM based, and hybrid approaches. The final submission adopts a hybrid strategy that combines linguistic normalization with semantic similarity,

while maintaining strict control over cluster growth and noise. All components are implemented using open source tools and can be executed locally without external API dependencies.

3.4.2. Data Handling and Preprocessing

Input data consists of a list of unique terms derived from the output of Subtask A. Terms are normalized by lowercasing and whitespace stripping. When enabled, linguistic preprocessing is performed using the `it_core_news_sm` SpaCy model to obtain Italian lemmas. Lemmatization is applied conservatively to avoid altering domain specific surface forms while still capturing inflectional variation.

3.4.3. Lemma Based Initial Grouping

As a first step, terms are grouped by their lemmatized forms. This provides an efficient baseline that captures morphological variants such as singular/plural alternations (e.g., *isola ecologica* vs. *isole ecologiche*). Lemma based grouping produces high precision but limited recall, as it does not account for lexical or semantic variation beyond morphology. For this reason, it is primarily used as an initialization step in the hybrid pipeline.

3.4.4. Embedding Based Semantic Clustering

To capture semantic similarity between non identical surface forms, we employ multilingual sentence embeddings computed with the `paraphrase-multilingual-MiniLM-L12-v2` model. Embeddings are computed for all terms and compared using cosine similarity. Several clustering algorithms are supported, including greedy threshold based clustering, DBSCAN, and hierarchical agglomerative clustering.

In the hybrid configuration, embedding similarity is used to merge lemma based clusters whose centroid embeddings exceed a similarity threshold. The similarity threshold is treated as a tunable hyperparameter and is optimized on the training set by maximizing BCubed F1 over a predefined range. The optimal threshold is stored and reused during development and test inference to ensure consistency.

3.4.5. Hybrid Clustering Strategy

The core clustering strategy combines lemma based grouping and embedding based refinement in a two stage pipeline. First, lemma based clusters are created to handle inflectional variants deterministically. Second, semantic similarity between cluster representatives is computed, and clusters are merged when their similarity exceeds the learned threshold. This approach balances precision and recall by exploiting complementary linguistic and semantic signals.

3.4.6. Strict Hybrid Pipeline and Anti Monster Cluster Controls

To further improve precision and prevent the formation of overly large and semantically incoherent clusters, we introduce a strict hybrid pipeline composed of four deterministic modules:

Acronym and Substring Solver A rule based preprocessing module identifies and locks obvious matches before semantic clustering. This includes acronyms and their expansions (e.g., *CCR* vs. *centro comunale di raccolta*) and strict substring matches after removing domain stopwords. Locked clusters are excluded from later merging steps to preserve high confidence relations.

Refined Embedding Clustering with Anti Lumping Logic Embedding based merging is performed using a stricter similarity threshold (0.90) and additional guardrails. Similarity scores are penalized when overlap is driven primarily by generic domain stopwords (e.g., *rifuti*, *raccolta*). Explicit antonym checks prevent merging terms that differ along critical semantic dimensions (e.g., *differenziato* vs. *indifferenziato*).

Component	Configuration
Embedding Model	paraphrase-multilingual-MiniLM-L12-v2
Similarity Metric	Cosine similarity
Clustering Strategy	Hybrid (lemma + embeddings)
Similarity Threshold	Optimized on training set
Strict Threshold (Anti lumping)	0.90
Monster Cluster Size Limit	Empirical (configurable)
Clustering Algorithms	Greedy, DBSCAN, HAC
LLM Verification	Disabled (official run)

Table 1

Hyperparameters and configuration for Subtask B clustering.

Subtask	Metric	System	Precision	Recall	F1
Subtask A	Micro	Baseline	0.497	0.559	0.526
		Proposed system	0.489	0.395	0.437
	Type	Baseline	0.435	0.508	0.469
		Proposed system	0.528	0.404	0.458
Subtask B	BCubed	Baseline	0.177	0.396	0.245
		Proposed system	0.390	0.333	0.359

Table 2

Official test-set results comparing Proposed system with the baseline for Subtask A (Automatic Term Extraction) and Subtask B (Term Variants Clustering).

Monster Cluster Breaker Clusters exceeding a predefined size threshold are identified as potential “monster clusters”. These clusters are recursively split using agglomerative clustering with cosine distance and average linkage. Internal cluster coherence is evaluated via average pairwise similarity, and low coherence groups are further decomposed into smaller clusters or singletons.

3.4.7. Training and Threshold Optimization

The cosine similarity threshold used for embedding based cluster merging was optimized on the training set by maximizing the BCubed F1 score over a predefined range of candidate values. The optimal threshold was found to be 0.85 and was subsequently fixed for all development and test experiments. This value is persisted and reused across runs to ensure consistent clustering behavior and full reproducibility. Such explicit threshold optimization allows the system to adapt to the domain specific distribution of term variants without relying on manual tuning.

3.4.8. Evaluation Protocol

Clustering performance is evaluated using BCubed Precision, Recall, and F1, following the official ATE-IT guidelines. Precision measures cluster purity, while recall reflects the completeness of concept coverage. Evaluation is performed on the development set during model selection and on the test set for final submission. All terms are guaranteed to be assigned to a cluster, with unmatched terms defaulting to singleton clusters.

4. Results

This section reports the official evaluation results of the system on the ATE-IT Shared Task test set. We compare our system against the official baseline provided by the task organizers for both subtasks.

Sentence	Gold Term(s)	System Output	Category
NUOVO CALENDARIO RACCOLTA RIFIUTI	calendario raccolta rifiuti	calendario raccolta rifiuti	True Positive
COMUNE DI AMATO	—	—	True Negative
Raccolte solo per i commercianti	—	raccolte	False Positive
libri, quaderni e album disegno	imballaggi in cartone	—	False Negative

Table 3

Representative qualitative examples for Subtask A.

4.1. Subtask A: Automatic Term Extraction

Table 2 reports the official test-set performance of the proposed system compared with the baseline. The results show that the system achieves competitive precision while trading off recall due to strict constraint-aware decoding and domain-specific filtering.

In Micro evaluation, the proposed system achieves strong precision but lower recall, reflecting a conservative extraction strategy. In Type evaluation, the system improves over the baseline in precision, indicating the ability to produce a cleaner and more coherent terminology inventory at the vocabulary level.

4.1.1. Observations

The extraction component is intentionally precision-oriented. Constraint-aware reconstruction, duplicate removal, and domain-specific filtering significantly reduce noise but also remove borderline valid terms, which lowers recall. This behavior is particularly visible for long multi-word expressions and syntactically complex phrases, where uncertainty accumulates across tokens.

Despite lower recall, the system produces highly consistent and terminologically coherent outputs. Administrative headers, institutional metadata, and non-domain fragments are consistently filtered out. This design choice favors high-quality term inventories suitable for downstream terminology management rather than exhaustive coverage.

Error analysis suggests that most remaining false negatives correspond to very long expressions or implicit domain terms that are difficult to detect through token-level prediction alone.

4.1.2. Qualitative Analysis

To complement the quantitative evaluation, we analyze representative predictions from the test set. Table 3 shows one example for each outcome category: true positive, true negative, false positive, and false negative.

These examples highlight the strengths and limitations of the proposed hybrid neural-symbolic approach. Domain-relevant expressions such as *calendario raccolta rifiuti* are correctly identified even in uppercase titles and non-sentential contexts. At the same time, occasional false positives emerge from generic administrative phrases, while false negatives mainly involve long or structurally complex expressions.

4.2. Subtask B: Term Variants Clustering

4.2.1. Observations

The hybrid and strict hybrid configurations consistently outperform single method baselines. Lemma based clustering yields very high precision but low recall, while embedding based clustering improves recall at the cost of increased noise. The proposed hybrid pipeline achieves a favorable balance by combining deterministic linguistic rules with semantic similarity and explicit anti lumping mechanisms. Error analysis reveals that most remaining errors arise from borderline cases involving abstract admin-

Concept	Gold Cluster Terms	Predicted Cluster Terms	Outcome
Abbandono dei rifiuti	abbandono; abbandono dei rifiuti; abbandono di materiali; abbandono irregolare dei rifiuti	abbandono; abbandono dei rifiuti; abbandono di materiali; abbandono irregolare dei rifiuti	Correct cluster
Calendario di raccolta	calendario; calendario raccolta rifiuti; calendario raccolta differenziata; calendario porta a porta	calendario; calendario raccolta; calendario raccolta rifiuti; calendario raccolta differenziata; calendario porta-a-porta	Correct cluster
Centri di raccolta	centro comunale di raccolta; centro di raccolta; isola ecologica; ecocentro; stazione ecologica	centro comunale di raccolta; centro di raccolta; isola ecologica; ecocentro; stazione ecologica; ccr; c.c.r.	Correct cluster
Servizi di gestione rifiuti	servizio di gestione dei rifiuti urbani; servizio di raccolta; servizi di raccolta	attivazione del servizio; attivazione del servizio di gestione dei rifiuti urbani; avvio del servizio integrato di gestione dei rifiuti urbani	Over grouping

Table 4

Qualitative comparison of gold and predicted clusters for Subtask B.

istrative phrases or subtle hypernymic relations, which remain challenging without deeper semantic modeling.

4.2.2. Qualitative Error Analysis

To complement BCubed based quantitative evaluation, we perform a qualitative analysis by comparing representative gold standard concept groupings with the clusters produced by our system. Table 4 reports illustrative examples highlighting correct cluster formation as well as remaining semantic over grouping phenomena.

The qualitative analysis confirms that the proposed hybrid clustering strategy is effective at capturing strong semantic equivalence relations, particularly for morphologically and lexically related variants. Concepts such as *abbandono dei rifiuti*, *calendario di raccolta*, and *centro di raccolta* are clustered correctly, demonstrating the effectiveness of lemma based initialization combined with embedding based refinement.

At the same time, the analysis reveals instances of semantic over grouping. In particular, expressions related to service activation (e.g., *attivazione del servizio*, *avvio del servizio integrato*) are grouped together with operational waste management services, despite representing procedural or administrative events rather than the service concepts themselves. This behavior highlights the difficulty of distinguishing closely related but conceptually distinct terms in an institutional domain characterized by dense lexical overlap.

Overall, remaining errors primarily involve subtle distinctions between administrative actions and operational concepts, suggesting that future improvements should focus on incorporating ontology aware or role sensitive semantic constraints.

5. Discussion

The experiments highlight several insights. First, neural sequence labeling alone is insufficient for high quality ATE in institutional domains; symbolic post processing is essential for controlling noise. Second, probability aware decoding provides a lightweight alternative to retraining for mitigating class imbalance effects. Third, clustering benefits from explicit mechanisms to prevent over merging, particularly in domains with dense lexical overlap.

Limitations include reduced recall due to conservative filtering and sensitivity of clustering performance to similarity thresholds. These trade offs are deliberate, favoring terminological precision over

coverage.

6. Conclusion

This paper presents the proposed system for the ATE-IT Shared Task at EVALITA 2026. By combining supervised transformer based modeling with domain-aware symbolic constraints and a hybrid clustering pipeline, the system achieves competitive and reproducible performance across both subtasks. Future work will explore class weighted training objectives, CRF-based decoding, and domain adapted embeddings to further improve recall and semantic resolution.

Acknowledgments

We thank the organizers of the ATE-IT Shared Task for providing the dataset and evaluation framework. We are also grateful to Prof. Paolo Torroni (University of Bologna) for his guidance and valuable discussions in the context of the Natural Language Processing course, which contributed to the development and refinement of this work. Finally, we acknowledge the open source NLP community for the tools and models that made this research possible.

Declaration on Generative AI

During the preparation of this work, the authors used generative AI tools (ChatGPT and Gemini) exclusively for writing assistance, grammar checking, and stylistic refinement. These tools were not used to generate scientific content, design the system, develop models, conduct experiments, or derive results or conclusions. All system implementations, experimental results, analyses, and interpretations are based on original work by the authors, who carefully reviewed and edited all AI-assisted text and take full responsibility for the content of this paper.

References

- [1] N. Cirillo, G. M. Di Nunzio, and F. Vezzani. *ATE-IT at EVALITA 2026: Overview of the Automatic Term Extraction Italian Testbed Task*. In *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*. CEUR.org, February 2026, Bari, Italy.
- [2] E. Amigó, J. Gonzalo, J. Artiles, and F. Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, 2009.
- [3] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29, 1990.
- [4] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 79–85, 1998.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 2019.
- [6] R. Wang, W. Liu, and C. McDonald. Neural attention models for sequence classification: Analysis and application to key term extraction. *arXiv preprint arXiv:1604.00077*, 2016.
- [7] M. Honnibal and I. Montani. spaCy: Industrial-strength Natural Language Processing in Python. Zenodo, 2017.
- [8] N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of EMNLP-IJCNLP*, 2019.

TrietNLP at ATE-IT: A Hybrid Pipeline for Italian Waste Management Terminology Analysis

Nguyen Minh Triet^{1,2,*}, Dang Van Thin^{1,2}

¹University of Information Technology, Ho Chi Minh City, Vietnam

²Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

This paper presents the approach of the TrietNLP team for the ATE-IT 2025 Shared Task on Automatic Term Extraction and Clustering in the domain of Municipal Solid Waste Management. The task involves two subtasks: Automatic Term Extraction (Subtask A) and Term Variants Clustering (Subtask B). For Subtask A, we propose a supervised neural architecture leveraging XLM-RoBERTa Large combined with a Conditional Random Field (CRF) layer to handle sequence labeling and boundary detection. For Subtask B, we employ a hybrid methodology that integrates rule-based pre-clustering for morphological variants with a Generative AI approach (Gemini 2.5) for semantic clustering. Our system utilizes incremental context injection to maintain consistency across batches. This report details the system architecture, preprocessing strategies, and the specific optimization techniques applied to address the lexical variability of Italian administrative texts.

Keywords

Term Extraction, Clustering, Natural Language Processing, XLM-RoBERTa, Generative AI, Waste Management

1. Introduction

The automatic identification and standardization of domain-specific terminology constitute a fundamental challenge in Natural Language Processing, particularly when applied to specialized administrative sectors. The ATE-IT 2025 shared task, organized within the EVALITA evaluation campaign [1], focuses on these challenges within the specific domain of Municipal Solid Waste Management. The data provided for this task comprises administrative documents, regulations, and official notices from Italian municipalities. This corpus presents significant linguistic hurdles, including high lexical variability, regional phrasings, abbreviations, and the necessity to distinguish between technical nomenclature and common language usage.

The shared task is divided into two interconnected subtasks designed to facilitate information standardization:

- **Subtask A: Automatic Term Extraction** is a sentence-level information extraction task. The objective is to automatically identify the boundaries of domain-specific terms, which may range from unigrams to complex multi-word expressions. A key challenge in this subtask involves handling nested terms to produce a flat list of technical spans.
- **Subtask B: Term Variants Clustering** functions as an unsupervised or semi-supervised clustering task. The goal is to group unorganized terms referring to the same operational concept into unified clusters. This requires the system to recognize inflectional variants, acronyms expansions, and true synonyms while avoiding the grouping of hierarchical concepts such as hypernyms.

In this paper, we present a comprehensive pipeline approach tailored to the specific constraints of the Italian waste management domain. Our contribution consists of two distinct strategies optimized for each subtask. For the extraction phase, we employ a neural sequence labeling architecture utilizing **XLM-RoBERTa** combined with a **Conditional Random Field** layer, implementing a longest-match-first strategy to resolve nested term conflicts. For the clustering phase, we propose a hybrid methodology

EVALITA 2025: Evaluation Campaign of Natural Language Processing and Speech Tools for Italian

*Corresponding author.

✉ 23521652@gm.uit.edu.vn (N. M. Triet); thindv@uit.edu.vn (D. V. Thin)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

that integrates rule-based logic with the semantic reasoning capabilities of Generative AI (specifically Gemini 2.5) to achieve high-precision grouping of synonyms and variants.

2. System Description

Our proposed system adopts a pipeline architecture that treats the two subtasks independently, employing a Neural Sequence Labeling approach for term extraction and a Hybrid (Rule-based + LLM) strategy for clustering.

2.1. Subtask A: Automatic Term Extraction

We model Subtask A as a token-level sequence labeling problem. Given the domain-specific nature of the dataset (waste management administrative acts), our approach focuses on robust context embedding and structural output validity.

2.1.1. Data Preprocessing and Labeling Strategy

The input data undergoes standard normalization, including Unicode NFKC normalization and lower-casing. We also remove specific noise characters (e.g., brackets [], ()) while preserving alphanumeric tokens.

A significant challenge in the provided dataset is the presence of "nested terms" (e.g., the term *"rifiuti urbani"* appearing inside *"trattamento rifiuti urbani"*). Standard sequence labeling models typically require a flat label structure. To address this, we implement a **Longest-Match-First** strategy during the creation of the gold-standard BIO (Beginning-Inside-Outside) labels:

- **Sorting:** All gold terms for a sentence are sorted by length in descending order.
- **Greedy Assignment:** We iterate through the sorted terms. If a term's span overlaps with an already labeled span, it is discarded. This prioritizes the longest maximal semantic units (e.g., keeping *"centro di raccolta"* and ignoring the nested *"raccolta"*).
- **Flattening:** This process effectively flattens the nested hierarchy into a single sequence of non-overlapping BIO tags, which is essential for training the CRF layer.

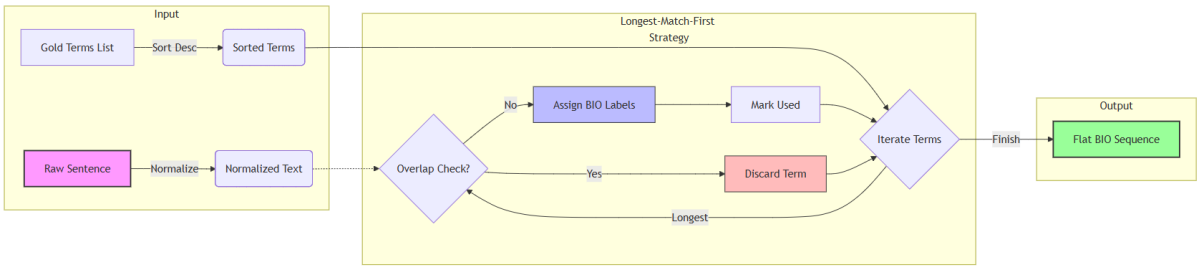


Figure 1: Illustration of the Longest-Match-First Labeling Strategy. The system prioritizes the longer span ("centro di raccolta") over the nested term ("raccolta"), resulting in a flat BIO tag sequence.

2.1.2. Model Architecture

The core architecture is a deep neural network composed of three main blocks:

1. **Backbone Encoder (XLM-RoBERTa):** We utilize xlm-roberta-large as the pre-trained encoder [2]. Its multilingual pre-training allows it to capture complex morphological and syntactic features of the Italian language effectively.

2. **Projection Layer (Highway-style):** Instead of feeding the raw encoder embeddings directly into the classifier, we introduce a projection block. This block consists of a Linear layer ($H \rightarrow H$), followed by a Tanh activation and Layer Normalization. This acts as a feature refinement stage, stabilizing the embeddings and reducing the variance of the pre-trained representations before classification.
3. **Decoding Layer (CRF):** To ensure the structural validity of the predicted tags, we employ a Conditional Random Field (CRF) layer [3]. Unlike a simple Softmax layer which predicts labels independently, the CRF models the transition probabilities between tags.
Implementation Detail: We explicitly constrain the transition matrix before training. Transitions that violate the BIO scheme (e.g., 'O' $\rightarrow$ 'I-TERM') are manually set to a large negative value ($-1e^4$) and frozen, forcing the model to learn only valid structural sequences.

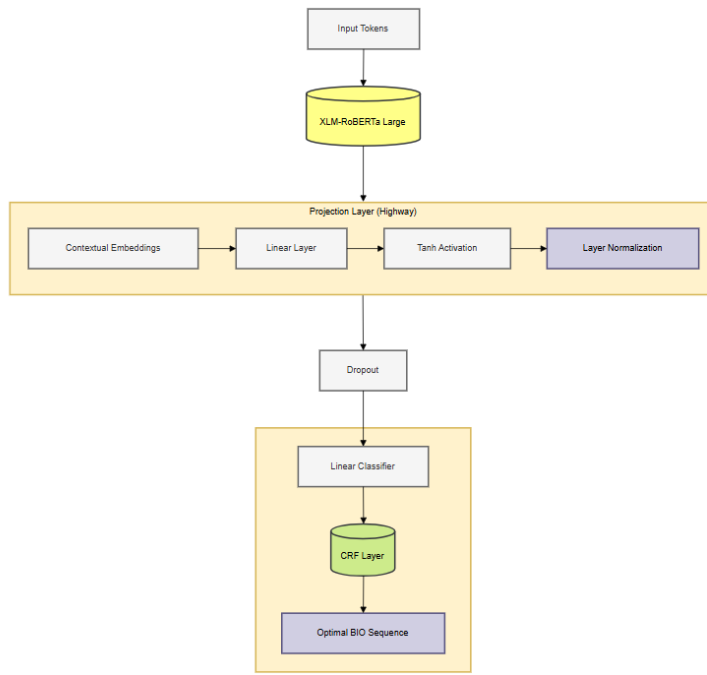


Figure 2: Model Architecture Diagram. The raw embeddings from XLM-R Large are refined via a Projection block (Linear+Tanh+LayerNorm) before passing to the CRF layer for final sequence decoding.

2.1.3. Training and Inference

The model was trained using the *Hugging Face Transformers* library with a batch size of 16 and a learning rate of $1e^{-5}$ for 6 epochs. During inference, we apply Viterbi Decoding via the CRF layer to obtain the globally optimal tag sequence. A final post-processing step extracts terms from the predicted spans, enforces lowercasing, and performs a final check to remove any duplicate or nested terms (keeping only the longest supersets) to strictly adhere to the Subtask A output format.

2.2. Subtask B: Term Variants Clustering

For the clustering of term variants, we developed a Hybrid Approach that combines deterministic rule-based filtering with the semantic reasoning capabilities of Generative AI. This ensures high precision for obvious morphological variants while leveraging LLMs for semantic synonyms.

2.2.1. Phase 1: Rule-based Pre-clustering

The first phase aims to reduce the search space and computational cost by grouping obvious morphological variants.

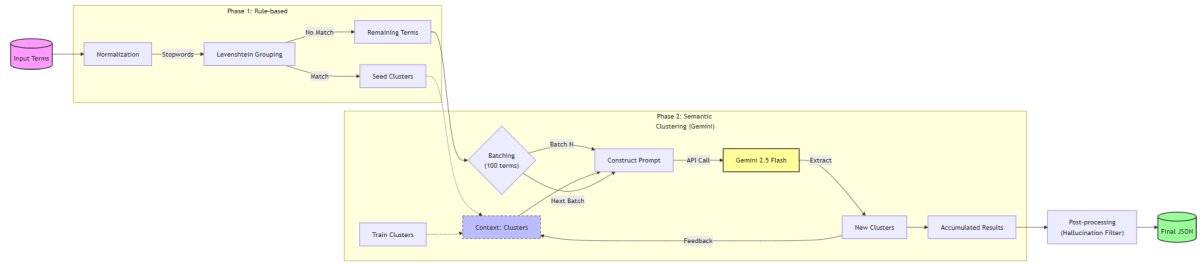


Figure 3: Overview of the Subtask B Pipeline: Input terms first pass through Rule-based Pre-clustering (Stop-words/Stemming) to form Seed Clusters, followed by an incremental Gemini LLM Loop for semantic grouping.

- **Normalization & Light Stemming:** Terms are cleaned by removing accents and punctuation. We apply a custom "light stemming" for Italian, normalizing vowel endings (e.g., substituting ending *-e* with *-a*, *-i* with *-o*) to handle simple gender/number variations.
- **Heuristic Grouping:** We filter out common Italian stopwords (e.g., *di*, *del*, *della*). Terms that reduce to the same normalized form and have a length difference ≤ 5 characters are grouped into "Seed Clusters". This effectively captures variants like *"isola ecologica"* vs *"isole ecologiche"* without LLM inference.

2.2.2. Phase 2: Semantic Clustering with Generative AI

Terms that remain unclustered or are only partially clustered are processed by Gemini 2.5 Flash [4]. This model was selected for its cost-efficiency and large context window, which is crucial for maintaining global consistency.

Prompt Engineering We designed a specific "System Prompt" that frames the LLM as a domain expert in Municipal Solid Waste Management. The prompt explicitly defines clustering constraints:

- **Positive Constraints:** Group acronyms with expansions (e.g., *"CCR"* = *"Centro Comunale di Raccolta"*) and true synonyms (e.g., *"Ecocentro"* = *"Isola ecologica"*).
- **Negative Constraints:** Strictly forbid grouping hypernyms with hyponyms (e.g., *"rifiuto"* $\neq$ *"rifiuto organico"*).

Incremental Clustering with Context Injection To handle the large volume of terms, we employ an incremental batch processing strategy (Batch size = 100):

1. The system initializes the context with high-confidence clusters derived from the Training set and Phase 1 rules.
2. For each new batch of unclustered terms, the prompt receives the **current state of all existing clusters**.
3. The model is instructed to: *"Insert the new terms into existing clusters when they are true synonyms, otherwise create new clusters."*

This "memory-augmented" loop ensures that a term in Batch N can be correctly clustered with a synonymous term processed in Batch 1, preventing the fragmentation of concepts across batches.

2.2.3. Post-processing

The LLM output is parsed from text to structured JSON. A final validation step filters out any "hallucinated" terms (terms generated by the LLM that were not in the original input list) and assigns any remaining isolated terms to singleton clusters to maximize recall.

3. Data and Experimental Setup

3.1. Dataset

The dataset used in this study is provided by the ATE-IT 2025 Shared Task organizers, specifically focusing on the domain of Italian Municipal Solid Waste Management. The corpus comprises administrative documents, regulations, and official notices, which are characterized by complex bureaucratic language and domain-specific terminology. The data is partitioned into three splits: Training, Development (Dev), and Test.

3.1.1. Data Preprocessing

To handle the linguistic variability inherent in the source text, we applied the following preprocessing pipeline:

- **Normalization:** We normalized all texts using the Unicode NFKC form to standardize characters.
- **Noise Removal:** Non-informative punctuation and noise characters (e.g., brackets, quotes) were removed, while alphanumeric tokens were preserved.
- **Lowercasing:** The entire corpus was converted to lowercase to mitigate vocabulary sparsity issues.

For **Subtask A**, the gold standard contained nested terms. To adapt this for sequence labeling models, we employed a "Longest-Match-First" strategy. This approach prioritizes longer terms over nested sub-terms to generate a flat BIO (Beginning-Inside-Outside) labeling scheme.

Table 1 summarizes the dataset statistics:

Split	Sentences	Annotated Terms
Training	2,308	2,218
Development	577	451
Test	1,142	-

Table 1

Statistics of the ATE-IT 2025 Dataset.

3.2. External Resources

Our system leverages the following external resources and pre-trained models:

- **XLM-RoBERTa Large:** We utilized the xlm-roberta-large model from Hugging Face [2] as the backbone for the term extraction task. This multilingual masked language model was pre-trained on 2.5TB of filtered CommonCrawl data.
- **Gemini 2.5 Flash[4]:** For the semantic clustering task, we utilized Google’s Gemini 2.5 Flash model accessed via the Generative AI API.
- **Libraries:** The implementation relies on the Transformers library for model fine-tuning and PyTorch-CRF for the Conditional Random Field layer.

3.3. Experimental Settings

3.3.1. Subtask A: Automatic Term Extraction

We modeled the term extraction task as a token classification problem. The architecture consists of an XLM-RoBERTa Large encoder followed by a Linear layer and a Conditional Random Field (CRF) layer to ensure the structural validity of the predicted BIO sequences.

The model was fine-tuned using the following hyperparameters:

- **Max Sequence Length:** 256 tokens.

- **Batch Size:** 16.
- **Learning Rate:** 1×10^{-5} .
- **Epochs:** 6.
- **Dropout:** 0.2.
- **Seed:** 36.

3.3.2. Subtask B: Term Variants Clustering

For the clustering task, we developed a hybrid pipeline combining rule-based heuristics with Generative AI:

- **Rule-based Pre-clustering:** A module that groups terms based on Levenshtein distance (threshold ≤ 5) and morphological suffix normalization to handle simple inflectional variants.
- **Generative AI Clustering:** The remaining unclustered terms were processed by Gemini 2.5 Flash. We used an incremental prompting strategy with a batch size of 100 terms to maintain context and consistency.

3.4. Evaluation Metrics

System performance is evaluated using the official metrics defined by the ATE-IT 2025 Shared Task organizers.

3.4.1. Subtask A Metrics

Performance for term extraction is measured using two complementary F1-scores: **Micro F1** and **Type F1** [5].

Micro F1 Score This metric evaluates the precision and recall across all sentences in the dataset, accounting for term frequency. Let TP_s , FP_s , and FN_s be the number of True Positives, False Positives, and False Negatives in sentence s , respectively. The micro-averaged Precision and Recall are defined as:

$$Precision_{micro} = \frac{\sum_{s \in D} TP_s}{\sum_{s \in D} (TP_s + FP_s)}, \quad Recall_{micro} = \frac{\sum_{s \in D} TP_s}{\sum_{s \in D} (TP_s + FN_s)}$$

The Micro F1 is the harmonic mean of these values:

$$F1_{micro} = \frac{2 \cdot Precision_{micro} \cdot Recall_{micro}}{Precision_{micro} + Recall_{micro}}$$

Type F1 Score This metric evaluates performance over the set of unique term types, disregarding their frequency in the corpus. Let TP , FP , and FN be the counts calculated on the set of unique extracted terms versus unique gold standard terms.

$$Precision = \frac{TP}{TP + FP}, \quad Recall = \frac{TP}{TP + FN}, \quad F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

3.4.2. Subtask B Metrics

Clustering quality is evaluated using the **BCubed F1 score** [6, 7]. BCubed calculates precision and recall at the item level and averages them across all items.

Let N_{pred} and N_{gold} be the number of elements in the predicted and gold clustering, respectively. Let $C(x)$ denote the predicted cluster containing element x , and $L(x)$ denote the gold cluster containing element x . For each element x , the item-level precision $P(x)$ and recall $R(x)$ are computed as:

$$P(x) = \frac{|\{y \in C(x) : L(y) = L(x)\}|}{|C(x)|}, \quad R(x) = \frac{|\{y \in L(x) : C(y) = C(x)\}|}{|L(x)|}$$

The global Precision and Recall are averages over all items in the predicted and gold sets, respectively:

$$Precision = \frac{1}{N_{pred}} \sum_{x=1}^{N_{pred}} P(x), \quad Recall = \frac{1}{N_{gold}} \sum_{x=1}^{N_{gold}} R(x)$$

Finally, the BCubed F1 score is calculated as:

$$F1_{BCubed} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

4. Results

In this section, we present the experimental results obtained by the **TrietNLP** team for the ATE-IT 2025 Shared Task. We report performance metrics on both the internal Development set and the official Test set as provided by the task organizers.

4.1. Subtask A: Automatic Term Extraction

For the Automatic Term Extraction task, performance is evaluated using Micro-averaged and Type-based Precision, Recall, and F1-scores.

4.1.1. Development Set Performance

During the development phase, we evaluated our proposed architecture (XLM-RoBERTa coupled with a Conditional Random Field layer) on the held-out development set. As shown in Table 2, the system achieved a Micro-F1 score of 0.718, demonstrating strong capability in identifying domain-specific terminology boundaries.

Table 2

Subtask A Results on the **Development Set** (TrietNLP).

Metric	Precision	Recall	F1-Score
Micro-Averaged	0.738	0.689	0.718
Type-Based	0.662	0.632	0.647

4.1.2. Official Test Set Results

Table 3 presents the official results on the blind Test set. **TrietNLP** ranked second overall (Top 2) in this subtask.

Our system significantly outperformed the official Baseline, achieving a Micro-F1 of **0.599** compared to the baseline’s 0.526. The inclusion of the projection layer and the CRF module proved essential for maintaining structural consistency in term prediction, allowing our model to remain competitive with the top-ranking system (Micro-F1 0.614).

4.2. Subtask B: Term Variants Clustering

For the Term Variants Clustering task, the evaluation utilizes Bcubed metrics, which are standard for clustering tasks where the number of clusters is not known a priori.

4.2.1. Development Set Performance

Our hybrid approach, combining rule-based pre-clustering with the incremental generative capabilities of Gemini 2.5, yielded exceptionally high results on the development set. Table 4 highlights a Bcubed F1-score of 0.985.

Table 3

Subtask A Official Results on the **Test Set**. Comparison between the Top 1 system, TrietNLP (Ours), and the Baseline.

Metric	System	Precision	Recall	F1-Score
Micro-Averaged	Top 1	0.656	0.577	0.614
	TrietNLP (Ours)	0.634	0.568	0.599
	Baseline	0.497	0.559	0.526
Type-Based	Top 1	0.645	0.529	0.581
	TrietNLP (Ours)	0.599	0.545	0.571
	Baseline	0.435	0.508	0.469

Table 4

Subtask B Results on the **Development Set** (TrietNLP).

Metric	Precision	Recall	F1-Score
Bcubed	0.975	0.944	0.985

4.2.2. Official Test Set Results

On the official Test set, **TrietNLP** achieved the highest performance, ranking as the **Top 1** system.

As detailed in Table 5, our system achieved a Bcubed F1-score of **0.441**, nearly doubling the performance of the Baseline (0.245) and surpassing the second-best system (0.359). This result validates the effectiveness of injecting context into the Large Language Model prompts to handle complex semantic relationships such as synonyms and acronyms in the waste management domain.

Table 5

Subtask B Official Results on the **Test Set**. Comparison between TrietNLP (Ours), the second-best system, and the Baseline.

System	Bcubed Precision	Bcubed Recall	Bcubed F1
TrietNLP (Ours)	0.528	0.378	0.441
Top 2	0.390	0.333	0.359
Baseline	0.177	0.396	0.245

4.3. Discussion

The discrepancy between Development and Test set results, particularly in Subtask B, suggests a significant shift in data distribution or term complexity in the official test data. However, the robust ranking of **TrietNLP** across both subtasks confirms that our methodology—specifically the combination of Transformer-based sequence labeling and LLM-augmented clustering—effectively generalizes better than the baseline approaches.

5. Discussion and Error Analysis

The performance of the TrietNLP system demonstrates the efficacy of separating the extraction and clustering tasks into distinct pipelines. However, an in-depth analysis of the prediction errors reveals specific linguistic and semantic challenges inherent to the domain of waste management in Italian.

5.1. Subtask A: Automatic Term Extraction

Despite achieving the second-best performance in the shared task, our error analysis based on the development set predictions highlights two primary categories of misclassification.

Boundary Detection and Determiner Inclusion. A recurrent error pattern involves the incorrect inclusion of Italian definite articles and prepositions within the term boundaries. As seen in the error report, the model frequently predicts terms such as *l'isola ecologica* or *dell'operatore ecologico*, whereas the Gold Standard expects the normalized forms *isola ecologica* and *operatore ecologico*. Although our preprocessing pipeline included Unicode normalization, the Conditional Random Field layer occasionally over-relied on local syntactic dependencies, failing to exclude the article preceding the noun phrase.

Long-tail and Nested Terminology. The model exhibits reduced recall for exceptionally long, multi-word expressions that resemble full clauses. For instance, the system failed to fully extract complex bureaucratic terms such as *servizio integrato gestione rifiuti – raccolta differenziata e servizi complementari*, often truncating them to shorter spans like *servizio integrato gestione rifiuti*. This suggests that while the XLM-RoBERTa encoder captures semantic meaning well, the fixed maximum sequence length and the rarity of such long n-grams in the training data hinder the model's ability to generalize to extreme cases.

5.2. Subtask B: Term Variants Clustering

Our hybrid approach, combining rule-based heuristics with the Gemini Large Language Model, achieved the top rank in the test phase. Nevertheless, specific challenges remain regarding semantic precision.

Hypernym vs. Hyponym Distinction. The primary challenge in unsupervised clustering is distinguishing between general categories and specific instances. Despite explicit instructions in the system prompt to separate distinct waste streams, the model occasionally struggled with the nuance between a general term (e.g., *rifiuti*) and its specific hyponyms (e.g., *rifiuti ingombranti*). This is a known limitation of current Large Language Models, which tend to associate terms based on semantic relatedness rather than strict ontological equivalence.

Incremental Learning and Semantic Drift. Our methodology employed an incremental batching strategy where previous clusters were fed back into the model context. While this improved consistency across batches, it introduced a risk of error propagation. If the model incorrectly grouped two distinct concepts in an early batch, subsequent batches would reinforce this error, leading to clusters that drifted from their original semantic definition.

6. Conclusion

In this report, we presented the TrietNLP system for the ATE-IT 2025 Shared Task. We approached the problem by decomposing it into two specialized sub-systems: a supervised neural sequence labeling architecture for term extraction and a hybrid neuro-symbolic approach for term clustering.

Our results validate this strategy. For Automatic Term Extraction, the combination of a multilingual encoder (XLM-RoBERTa) with a structured prediction layer (CRF) proved robust, capturing the majority of domain-specific terminology. For Term Variants Clustering, leveraging the reasoning capabilities of Large Language Models allowed us to handle complex synonyms and acronyms that simple string-matching algorithms miss, resulting in state-of-the-art performance on the test set.

Future Work. To address the limitations identified in our analysis, future developments will focus on two key areas:

- **Adversarial Training for Boundary Precision:** To mitigate the inclusion of articles and prepositions in Subtask A, we propose implementing adversarial training techniques. By deliberately exposing the model to perturbed inputs where articles are masked or modified, we aim to force the encoder to focus solely on the semantic core of the term.

- **Fine-tuning Small Language Models:** For Subtask B, replacing the generic API-based Large Language Model with a smaller, domain-finetuned model (e.g., Llama 3 or Mistral) could reduce latency and costs. Fine-tuning on specific ontology datasets would also likely reduce hallucination rates and improve the distinction between hypernyms and hyponyms.

Acknowledgments

We would like to thank the University of Information Technology, Ho Chi Minh City, Vietnam (UIT VNUHCM) for the support and resources provided during this research.

Declaration on Generative AI

During the preparation of this work, the authors used Google Gemini in order to: check for grammar and spelling errors. After using this tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication's content.

References

- [1] N. Cirillo, G. M. Di Nunzio, F. Vezzani, ATE-IT at EVALITA 2026: Overview of the Automatic Term Extraction Italian Testbed Task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 8440–8451. URL: <https://aclanthology.org/2020.acl-main.747>. doi:10.18653/v1/2020.acl-main.747.
- [3] J. Lafferty, A. McCallum, F. C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proceedings of the eighteenth international conference on machine learning (ICML), 2001, pp. 282–289.
- [4] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al., Gemini: a family of highly capable multimodal models, arXiv preprint arXiv:2312.11805 (2023).
- [5] R. Verborgh, M. Röder, R. Usbeck, A.-C. Ngonga Ngomo, GERBIL – Benchmarking Named Entity Recognition and Linking Consistently, Semantic Web 9 (2018) 605–625. URL: <https://doi.org/10.3233/SW-170286>. doi:10.3233/SW-170286.
- [6] A. Bagga, B. Baldwin, Entity-based cross-document coreferencing using the vector space model, in: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (COLING-ACL'98), Montreal, 1998, pp. 79–85.
- [7] E. Amigó, J. Gonzalo, J. Artiles, F. Verdejo, A Comparison of Extrinsic Clustering Evaluation Metrics Based on Formal Constraints, Information Retrieval 12 (2009) 461–486.

VVTE at ATE-IT: From Candidates to Terms: Hybrid Italian ATE with Dependency Heuristics, Gemini, and Random Forest Filtering

Valentine G. L. Vandervoort^{1,*}, Giorgio Maria Di Nunzio¹

¹Department of Information Engineering, University of Padova, Via Gradenigo 6/b, 35131 Padova, Italy

Abstract

Automatic Term Extraction (ATE) aims to identify domain-specific lexical units, including single- and multi-word expressions, that encode the conceptual structure of specialized fields. This paper presents our submission to Subtask A (Term Extraction) of the ATE-IT shared task [1] at EVALITA 2026 [2], the first large-scale evaluation campaign dedicated to Italian ATE, focused on institutional texts in the municipal waste-management domain. To cope with sparse terminology, rich Italian morphology, and complex administrative multi-word terms, we adopt a hybrid pipeline that prioritizes recall and then progressively filters candidates. We first generate a broad pool of candidate terms using dependency-based heuristics with spaCy and zero-shot term identification via Google Gemini. We then apply normalization and task-specific constraints and refine predictions with a Random Forest classifier trained on a custom feature set capturing morphological patterns, domain-keyword density, and semantic similarity signals. We report official results on the test set and provide an analysis of typical errors, highlighting the benefits and limitations of combining linguistic heuristics, generative models, and supervised learning for Italian institutional automatic term extraction.

Keywords

Automatic Term Extraction, ATE-IT, Linguistic features, Waste management

1. Introduction

Automatic Term Extraction (ATE) is a core task in Natural Language Processing (NLP) and computational linguistics, designed to identify domain-specific terms that represent key concepts within a specialized field of knowledge [3, 4, 5]. Unlike Named Entity Recognition (NER), which focuses on identifying unique referents such as specific individuals, organizations, or geographic locations, ATE aims to extract lexical units, both single-word and multi-word expressions, that constitute the conceptual framework of a domain. The outputs of ATE systems are vital for numerous downstream applications, including the construction of ontologies, enhancement of knowledge graphs, machine translation, and the domain adaptation of Large Language Models (LLMs).

This paper describes our participation in the ATE-IT [1] (Automatic Term Extraction – Italian Testbed) shared task, organized within the framework of EVALITA 2026 [2]. EVALITA is the main evaluation campaign for NLP and speech technologies for the Italian language [?]. The ATE-IT task represents an important step toward systematic evaluation for Italian ATE, with a focus on the institutional domain of waste management. This domain presents significant linguistic challenges, including a high density of derived terms (e.g., “*ecodizionario*”), technical abbreviations (e.g., “*TARI*”, “*RAEE*”), and complex, multi-word administrative terms (e.g., “*raccolta porta a porta*”).

The ATE-IT campaign is structured into two distinct subtasks: Term Extraction (Subtask A) and Term Variants Clustering (Subtask B). Our contribution focuses exclusively on Subtask A, which requires the identification of domain-relevant terms, including nouns, verbs, and adjectives, from a specialized corpus.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ valentinegabriel@unipd.it (V. G. L. Vandervoort); giorgiomaria.dinunzio@unipd.it (G. M. Di Nunzio)

🌐 <https://valouvalouvalou.github.io/> (V. G. L. Vandervoort)

🆔 0000-0001-7116-9338 (G. M. Di Nunzio)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

To address the inherent difficulty of identifying sparse, domain-specific terms in Italian, we propose a hybrid pipeline that combines linguistic heuristics, generative artificial intelligence, and supervised machine learning. Our approach first maximizes recall by extracting a broad set of candidates using spaCy for dependency parsing [?] and Google Gemini for zero-shot term identification. This candidate list is subsequently refined through a rigorous filtering process and a Random Forest classifier [?]. The classifier is trained on a custom feature set including morphological patterns, domain-keyword density, and semantic similarity to distinguish valid domain terms from general language.

In the following sections, we provide a comprehensive breakdown of our system architecture, followed by an analysis of our performance on the official test set and a discussion of the implications of our hybrid approach.

2. Description of the System

The proposed system follows a hybrid architecture designed to achieve high recall in the initial phase, followed by a precision-oriented filtering stage. The pipeline integrates rule-based linguistic analysis, generative AI, and supervised machine learning.

2.1. Preprocessing and Candidate Extraction

The first stage of the pipeline aims to identify a broad set of potential terms (candidates) from the raw text without immediate domain-specific filtering. This ensures that rare or complex terms are not overlooked.

Linguistic Parsing. I utilize the `it_core_news_lg` model from *spaCy* to perform tokenization, part-of-speech (POS) tagging, and dependency parsing. To handle complex sentence structures, each sentence is segmented using punctuation anchors (e.g., semicolons, question marks, and commas) while preserving the integrity of segments that lack these separators.

Multi-source Extraction. To capture a diverse range of terminological units, we employ several concurrent extraction strategies:

- **Syntactic Subtrees:** We extract subtrees anchored in nouns, subjects, verbs, and adjectival phrases.
- **Noun Chunks:** We leverage *spaCy*'s native noun chunking to identify base nominal groups.
- **Technical Abbreviations:** A specialized module extracts technical acronyms, filtering out common Italian stopwords and purely numerical strings.
- **Linguistic Patterns:** We specifically target prepositional phrases, compound nouns, appositions, and conjunction chains to capture complex multi-word terms.

LLM-Augmented Extraction. In addition to rule-based methods, we incorporate Google Gemini to identify candidates. By processing each sentence individually, the model leverages its internal knowledge to suggest terms that might escape rigid syntactic patterns, providing a robust complement to the heuristic-based extraction.

Initial Refinement. All extracted candidates undergo a cleaning process where leading and trailing stopwords are removed. This ensures that the boundaries of the candidates are linguistically sound before entering the validation phase.

2.2. Validation and Feature Engineering

Once the initial set of candidates is gathered, we apply a series of filters and transformation steps to prepare the data for the classification model. This stage is designed to eliminate noise and quantify the characteristics of each term.

Structural Filtering. We first perform a preliminary validation of the candidate's form. Any term containing digits, special characters (with the exception of hyphens and commas), or patterns resembling

dates is automatically rejected. Furthermore, we discard single-token candidates that appear in the Italian stopword list to ensure that only potentially meaningful units remain.

Feature Extraction. For each remaining candidate, we compute a comprehensive set of features that capture its morphological, lexical, and contextual properties:

- **Morphological features:** We calculate the word count, character length, and the ratio of vowels to consonants. We also check for the presence of typical Italian suffixes and prefixes that often characterize technical terminology.
- **Domain-specific features:** Using a pre-compiled list of domain keywords, we determine the number and percentage of keywords within the term. We also verify whether the candidate or its components appeared in the training dataset.
- **Contextual and Semantic features:** We record the term's relative position in the sentence and its internal structure (e.g., whether it is a compound noun). Additionally, We calculate the ratio of internal stopwords and the semantic similarity of the candidate to validated terms from the training set.

Normalization. Before passing these features to the classifier, we normalize the terms to ensure consistency across the dataset. This step is vital for the model to generalize effectively across different morphological variants of the same concept.

2.3. Supervised Classification and Final Selection

To distinguish genuine waste management terminology from general language candidates, we implemented a supervised learning approach designed to prioritize domain relevance.

Model Training. We trained a Random Forest classifier using the provided training dataset. To do this, we constructed a balanced dataframe consisting of positive examples (validated terms from the corpus) and negative examples (general language words and irrelevant extractions). To address the inherent sparsity of technical terms, we configured the model with a specific weight ratio, assigning three times more importance to the positive class (`class_weight=1: 3`). This weighting strategy is crucial for ensuring the model remains sensitive to domain-specific terminology despite the prevalence of common vocabulary.

Decision Process. During the inference phase, we submit each candidate to the trained model, along with its vector of morphological and contextual features. To maximize precision and minimize "noise" from false positives, we apply a strict dual-condition filter. A candidate is only accepted as a valid term if:

1. The model predicts it as a domain-relevant term (Class 1).
2. The prediction probability exceeds a confidence threshold of 0.7.

Output Generation. Finally, we reconstruct the definitive dictionary of identified terms. We take particular care to ensure the output data structure matches the input format exactly, maintaining the required alignment for the EVALITA evaluation scripts. The final results are exported into a structured JSON file, formatted for official submission.

3. Results

In this section, we present the official results obtained on the ATE-IT test set after processing the data through my hybrid pipeline. The performance is evaluated using the official metrics provided by the EVALITA 2026 organizers, focusing on both Micro (instance-level) and Type (unique term-level) scores.

Quantitative Performance. Table 1 summarizes the comparison between the official ATE-IT baseline and our system's output.

Analysis of the Precision-Recall Trade-off. The most striking result is the significant leap in Type Precision, where our system achieved 0.707, substantially outperforming the baseline's 0.435. This

System	M-Prec	M-Rec	M-F1	T-Prec	T-Rec	T-F1
Baseline	0.497	0.559	0.526	0.435	0.508	0.469
My System	0.364	0.473	0.411	0.707	0.262	0.382

Table 1

Comparison of official baseline results and my system performance on the test set.

outcome reflects a deliberate architectural choice: we prioritized high-confidence extractions over broad coverage. By setting a strict probability threshold of 0.7 and utilizing multi-stage filtering, we ensured that the majority of the unique terms identified were indeed valid and highly relevant to the waste management domain.

However, this conservative strategy resulted in a significant drop in Type Recall (0.262). Upon internal review of the output, it appears that while the extracted terms are highly accurate, they consist mostly of single-word or short multi-word units (maximum 3 words). The pipeline struggled to capture longer, more complex administrative expressions, leading to a high number of false negatives. This suggests that while my Random Forest classifier is highly effective at validating core domain vocabulary, the initial candidate extraction or the subsequent filtering was perhaps too restrictive for complex compositional terms.

Micro-Level Performance Analysis. Regarding instance-level performance, our system achieved a Micro Precision of 0.364 and a Micro Recall of 0.473. While these figures remain below the official baseline, the discrepancy between the Micro and Type metrics offers a deeper insight into the pipeline’s behavior. The gap between the high Type Precision (0.707) and the lower Micro Precision (0.364) suggests that while we successfully identify a correct unique concept once, the system lacks consistency in extracting every occurrence of that concept across the entire corpus. A term might be correctly validated in a favorable syntactic context but rejected elsewhere if the local features or the Gemini confidence score fall below the 0.7 threshold.

Furthermore, the fact that Micro Recall (0.473) is significantly higher than Type Recall (0.262) indicates that our system tends to reliably capture a core set of frequent, well-defined terms while missing the “long tail” of less frequent terminology. This confirms that the Random Forest classifier, coupled with the strict filtering rules, favored statistical safety. By prioritizing terms with strong morphological and keyword-based signatures the system ensured that the extracted output was qualitatively relevant to the waste management domain, even at the cost of exhaustive coverage and instance-level consistency.

4. Discussion

The evaluation results provide a nuanced view of the hybrid pipeline’s capabilities. While the system achieved an exceptional leap in Type Precision compared to the baseline, the trade-off in recall and the difficulty in capturing complex expressions offer important insights into the nature of the task and the limitations of the chosen architecture.

4.1. Successes and Model Selection

During the development phase, we experimented with several classification architectures. The Random Forest Classifier consistently proved to be the most stable and effective model on the training dataset, leading to its selection for the final system. The inclusion of *class_weight* adjustments was particularly effective in forcing the model to recognize domain-specific vocabulary that might otherwise have been overshadowed by general language tokens. The primary success of this approach is its ability to identify high-confidence, unique terms within the waste management domain. Our final system achieved a Type Precision of 0.707, which indicates a very high rate of true positives among the unique terms extracted.

However, this result must be interpreted with caution. While it demonstrates a high level of accuracy in identifying valid terminology, the score is also a mathematical consequence of a very low Type Recall (0.262). Because the system was highly selective and extracted a relatively small number of

unique terms, the pool of candidates for which precision was calculated was limited. Nevertheless, the score remains a positive indicator of the system’s ability to maintain a clean output, largely free from non-terminological noise.

4.2. Experiments with LLM Filtering

In one iteration of the pipeline, we integrated the Grok LLM via API to act as an initial filter immediately following the first extraction phase. The goal was to use the LLM to prune irrelevant candidates before they reached the feature engineering and classification stages. Interestingly, this additional layer of generative filtering resulted in negligible changes to the final evaluation scores. This suggests that the heuristic-based extraction and the subsequent Random Forest classifier were already capturing the most salient features of the data.

4.3. Analysis of Limitations and Short-term Bias

The main challenge encountered was a significant “short-term bias,” with the output consisting mostly of single words or very short multi-word units. We attribute this to two factors:

- **Classifier Rigidity:** The Random Forest model appears to have been the primary bottleneck for longer expressions. It likely penalized complex strings whose statistical features, such as stopword ratios or character lengths, deviated from the more frequent, shorter terms found in the training data.
- **Structural Fragmentation:** The aggressive segmentation strategy used to handle complex sentences may have inadvertently severed the continuity of longer institutional phrases, preventing the model from ever seeing the full multi-word candidate.

4.4. Future Work

To bridge the gap between precision and recall, future work will shift focus toward finding a more balanced equilibrium. We intend to move away from a precision-only focus and instead prioritize improving the Recall score. This will involve relaxing the current strict filtering rules and lowering the probability threshold to capture a wider variety of terms that were previously overlooked. Additionally, we plan to refine the extraction phase to better preserve the integrity of complex noun phrases and experiment with transformer-based embeddings to help the system generalize better to rare, long-tail terminology that the current model failed to capture.

Declaration on Generative AI

During the preparation of this work, the author used OpenAI GPT-5.2 and Google Gemini in order to perform grammar and spelling checks, as well as to assist with text reformulation for clarity and flow. After using these tools, the author reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

- [1] N. Cirillo, G. M. Di Nunzio, F. Vezzani, Ate-it at evalita 2026: Overview of the automatic term extraction italian testbed task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [3] K. Kageura, B. Umino, Methods of automatic term recognition: A review, Terminology. International Journal of Theoretical and Applied Issues in Specialized Communication 3 (1996) 259–289. doi:10.1075/term.3.2.03kag.
- [4] K. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms: the C-value/NC-value method, International Journal on Digital Libraries 3 (2000) 115–130. doi:10.1007/s007999900023.
- [5] Y. Chun, M. Kim, D. Kim, C. Park, H. Lim, Enhancing Automatic Term Extraction with Large Language Models via Syntactic Retrieval, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics: ACL 2025, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 9916–9926. doi:10.18653/v1/2025.findings-acl.516.

PFB at EVALITA 2026: Overview of the Prometeia Financial Benchmark

Alessandro P. Bardelli^{2,†}, Tolga Çekiç^{3,†}, Irem Demirtaş^{3,†}, Michele Filannino^{2,†},
Simona Scala^{1,*,†}, Andrea Galassi^{4,†}, Gianmarco Pappacoda^{4,†} and Paolo Torroni^{4,†}

¹Prometeia, Piazza Trento e Trieste 3, 40137, Bologna, Italy

²Prometeia, Via Brera, 18, 20121, Milan, Italy

³Prometeia, River Plaza, Kat 19 Büyükdere Caddesi Bahar Sokak No. 13, 34394, Levent | Istanbul | Turkey

⁴Università di Bologna, Dipartimento Informatica - Scienza e Ingegneria, Viale del Risorgimento 2, 40136, Bologna, Italy

Abstract

The Prometeia Financial Benchmark (PFB) is the EVALITA 2026 shared task on finance questions across 3 languages: Italian, English, and Turkish, and 3 difficulty levels: easy, medium, and hard. The challenge is organized in two subtasks, one on Italian data and one on all three languages. For each subtask, we have received 2 submissions. Our main takeaways are that no significant performance differences stand out across languages and difficulty levels, and that PFB appears to be a challenging benchmark for models smaller than 3B, whereas 20B models already reach an overall accuracy of 90%.

Keywords

Large Language Models, Finance NLP, Multiple-choice QA, Multilingual benchmark, Prometeia Financial Benchmark

1. Introduction

Language Models (LMs) are increasingly adopted as general-purpose components for knowledge-intensive workflows, and finance is a natural target for their use [1, 2]. Finance-related language and reasoning, however, impose stricter requirements than many general-purpose benchmarks capture: terminology is specialized, concepts are tightly interdependent, and answers that appear plausible at the surface level may still be incorrect. Recent evidence highlights reliability limitations in finance-specific settings, including hallucination-related issues [3], motivating domain-targeted evaluation protocols.

This paper presents the *Prometeia Financial Benchmark* (PFB) shared task, organized as a track within EVALITA26 [4], to benchmark LMs on finance-domain multiple-choice question answering (MCQA). The task is based on a curated dataset of 1,500 questions released in aligned English, Italian, and Turkish versions, and is structured into an Italian-only and a multilingual subtrack. We describe the task definition, dataset construction, and evaluation protocol, and provide an overview of participant submissions and results.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

†These authors contributed equally.

✉ alessandro.bardelli@prometeia.com (A. P. Bardelli); tolga.cekic@prometeia.com (T. Çekiç); irem.demirtas@prometeia.com (I. Demirtaş); michele.filannino@prometeia.com (M. Filannino); simona.scala@prometeia.com (S. Scala); a.galassi@unibo.it (A. Galassi); gianmarco.pappacoda@unibo.it (G. Pappacoda); p.torroni@unibo.it (P. Torroni)

ORCID 0009-0009-7670-5668 (A. P. Bardelli); 0009-0008-2061-5893 (T. Çekiç); 0009-0007-5586-9773 (I. Demirtaş); 0000-0001-8208-2238 (M. Filannino); 0009-0003-5324-5466 (S. Scala); 0000-0001-9711-7042 (A. Galassi); 0009-0001-6609-4156 (G. Pappacoda); 0000-0002-9253-8638 (P. Torroni)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Dedicated to the memory of Annalina Caputo.

2. Motivation

Finance is a high-stakes domain in which language technology is expected to support decision-making, compliance-driven workflows, and knowledge access over complex documentation. Although modern LMs perform well on broad benchmarks, their fluency can mask brittle understanding and overconfident errors [5]. In finance, such failures are especially problematic: incorrect statements may propagate into reports, internal procedures, or user-facing advisory contexts, increasing operational and financial risk [3, 6]. A dedicated benchmark is therefore needed to quantify reliability on domain content rather than generic linguistic ability.

Many finance tasks require distinguishing between closely related concepts, interpreting definitions precisely, and selecting the only option that is fully consistent with the question context. MCQA provides a controlled evaluation setting: the decision space is explicit, distractors can be designed to be plausible within the domain, and scoring is unambiguous. This helps separate genuine domain comprehension from persuasive surface generation.

Multilinguality is an additional practical driver. Financial institutions operate across markets, and key materials are often consumed in local languages, not only in English. Yet most available benchmarks remain predominantly English-centric, making it difficult to disentangle domain effects from language effects. Multilingual, aligned resources enable controlled cross-lingual comparisons and support analyses of robustness, transfer, and language-specific terminology handling [7, 8, 9, 10].

Finally, the shared-task format provides a transparent and reproducible comparison framework. It complements prior work on financial LLM evaluation and benchmarking [11, 12, 13] by fixing data, protocol, and metrics, while encouraging diverse modelling approaches and facilitating systematic error analysis.

3. Definition of the task

The shared task assesses a model’s ability to *understand and reason over finance-domain content* in a controlled MCQA setting. Given a question and five candidate answers, systems must return the option that best answers the question. The task is open with respect to modelling choices: participants were free to submit systems based on different paradigms (e.g., encoder-only or decoder-only LMs, instruction-tuned LLMs, prompted systems) and to rely on either open or proprietary models, including both “small” and “large” language models.

For each test instance, participants submit a single answer choice in {A,B,C,D,E}. Systems may optionally provide short textual justifications. These explanations are not used for ranking, but they may support qualitative analyses (e.g., recurring failure modes, reasoning patterns, and domain-specific errors).

The shared task is organized into two subtracks. The first is an **Italian-only** track, which constitutes the primary focus within EVALITA. The second is a **multilingual** track covering Italian, English, and Turkish, designed to enable controlled comparisons across aligned instances. Participants may submit to one or both subtracks.

For this task, participating teams were required to submit their primary run along with information about the model architecture, inference strategy, and related implementation specifics. Additionally, each team was allowed to submit up to four secondary runs.

3.1. Dataset

The task is based on the *Prometeia Financial Benchmark* (PFB) [14], a curated collection of 1,500 finance-domain items released in three aligned languages (English, Italian, and Turkish). Questions were derived from heterogeneous finance-related sources (e.g., reports, papers, and regulatory texts) and curated with expert review to ensure domain relevance, internal consistency, and clarity.

Each instance is identified by a language-invariant `custom_id`, enabling one-to-one alignment across the three languages. Provenance is captured by a `category` label indicating the source family. Items

Table 1

Core fields in the PFB dataset release

Field	Description
<code>custom_id</code>	Unique identifier, shared across languages for alignment.
<code>question</code>	Question stem.
<code>choiceA-choiceE</code>	Five candidate answers.
<code>correct_answer</code>	Gold option in {A,B,C,D,E}.
<code>difficulty_level</code>	Categorical difficulty in {easy, medium, hard}.

follow a standard MCQA schema with the stem (`question`), answer options (`choiceA-choiceE`), and the gold label `correct_answer` in {A,B,C,D,E}. In addition, the dataset provides a categorical indicator, `difficulty_level` (easy, medium, hard).

Table 2

Distribution of correct choices across labels in PFB dataset

Label	A	B	C	D	E
Number	402	438	424	237	0
Percentage	26.8	29.1	28.3	15.8	0

Table 3

Distribution of difficulty in PFB dataset

Difficulty	Easy	Medium	Hard
Number	248	1155	98
Percentage	16.6	76.9	6.5

Dataset construction followed a two-stage process: a large pool of candidate questions was first produced from the selected sources: financial texts (financial books, and regulatory texts), academic papers and financial reports (Table 4). Then iteratively filtered and refined through expert review. Quality control relied on structured assessment signals recorded at the item level—including *soundness*, *ambiguity*, *factuality*, and *relevance*—complemented by free-text notes. These signals guided revisions of stems and distractors, removal of problematic items, and documentation of corner cases. The dataset was originally constructed with four answer options (A–D), each containing exactly one correct response. Later, a fifth option (E), labeled as “None of the above” (and its counterparts in Italian and Turkish datasets) was appended to all questions. This option was intentionally designed to be incorrect in every case and serves as a controlled distractor. Its inclusion enabled the evaluation of the models’ susceptibility to uncertainty and their tendency to select generalized options.

Table 4

Distribution of question categories in PFB dataset

Category	Financial Texts	Financial Papers	Financial Reports
Number	327	339	335
Percentage	32.6	33.9	33.5

The benchmark is distributed in three languages with instance-level alignment. The Italian and Turkish versions were generated via automatic translation (DeepL Enterprise) and subsequently manually post-edited to correct domain terminology, preserve the semantics of both stems and answer options, and avoid language-specific artifacts that could inadvertently cue the correct answer. For the

shared-task workflow, PFB is released with a public development split and a held-out test split used for final evaluation; all splits preserve the `custom_id` alignment across languages. The dataset has two splits: 500 questions have been published for examples and 1001 questions have been used for testing. The splits are the same across all three languages.

In Table 2, we show the distribution of the correct choices, in Table 3 the distribution of difficulty levels and in Table 4 the distribution of question categories.

3.2. Evaluation Measures

Systems are evaluated using *accuracy*, i.e., the fraction of instances for which the selected option matches the gold label to be chosen among the available options. Given a test set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, where $y_i \in \{A, B, C, D, E\}$ is the correct option and $\hat{y}_i$ is the system prediction, the primary score is:

$$\text{Acc}(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[\hat{y}_i = y_i]. \quad (1)$$

In addition to overall accuracy on the full test set, we also report accuracy on predefined *difficulty-based subsets* derived from the item difficulty labels (e.g., easy vs. hard), to support diagnostic comparisons across systems.

4. Results

4.1. Baselines

As baselines, we use a similarity-based approach based on an encoder model and two small-sized LLMs:

- **Similarity** (0.5B): we select the answer as the one most similar to the question. The similarity score is computed using a Sentence-BERT model [15]: *distiluse-base-multilingual-cased-v1*.¹
- **Qwen** (1.7B) [16]: we use *Qwen3-1.7B*² (instruction tuned).
- **Llama** (3B) [17]: we use *Llama-3.2-3B-Instruct*.³

For the LLMs, our prompt contains the question, the list of possible answer, and their labels in the following format:

```
Question: [question]
A: [choiceA]
B: [choiceB]
C: [choiceC]
D: [choiceD]
E: [choiceE]
Answer:
```

All the baselines have been pre-trained, to some extent, in the three languages considered in this task: Italian, English, Turkish.

4.2. Systems Overview

We received two submissions. The **UNITOR** system by Borazio et al. [18] proposes an agentic architecture without any fine-tuning or domain adaptation that follows a three-stage orchestration pipeline. The first stage is semantic routing, used to distinguish between three different reasoning strategies:

¹<https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1>

²<https://huggingface.co/Qwen/Qwen3-1.7B>

³<https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct>

quantitative analysis, Boolean fact-checking, and knowledge-based research reasoning. Following routing, a specialized reasoning phase instantiates multiple parallel inference threads aimed at exploring parallel reasoning paths. Finally, an aggregation stage uses majority voting to select the most stable answer under sampling, following [19]. An iterative refinement step may fire if the consensus is weak. During refinement, the reasoning modules are re-invoked with a constrained version of the original prompt, in which low-support options are explicitly masked. The authors evaluate this architecture on three open-weight models: LLaMA 3.1 (8B) as a lightweight baseline, GPT-OSS-20B (21B) as the primary reference model, and DeepSeek v3.1 (671B) as a large upper bound. The top performer in the validation set used for the official submission is DeepSeek 3.1 for English, and GPT-OSS-20B for Italian and Turkish, achieving about 0.88 average accuracy across all languages.

The AMSN system by Mohammadabad and Nazarmohsenifakori [20] is the result of analysis of three different approaches: fine-tuning transformer-based models for question-answering, LLM prompting, and a hybrid approach. The first approach leverages mDeBERTa-v3. The second one is used to evaluate GPT-4o and GPT-5 with several prompting strategies. The hybrid approach uses a specialized trained model to detect and correct GPT-4o’s mistakes. The top performer in the validation set used for the official submission is GPT-5, which achieves nearly 0.90 average accuracy across all languages.

4.3. Experimental Results

Table 5

Results on the two subtasks

	Subtask 1	Subtask 2
AMSN [20]	0.91	0.90
UNITOR [18]	0.88	0.88
Llama [17]	0.37	0.37
Qwen [16]	0.33	0.29
Similarity [15]	0.21	0.20

Table 6

Breakdown of the results according to languages and difficulty of the instances (all, hard, medium, and easy)

Language	IT				EN				TR				ALL			
	A	H	M	E	A	H	M	E	A	H	M	E	A	H	M	E
AMSN [20]	0.91	0.92	0.92	0.86	0.89	0.88	0.90	0.84	0.88	0.88	0.89	0.88	0.90	0.89	0.90	0.86
UNITOR [18]	0.88	0.84	0.88	0.89	0.89	0.91	0.90	0.88	0.88	0.88	0.88	0.89	0.88	0.88	0.89	0.89
Llama [17]	0.37	0.41	0.36	0.38	0.43	0.48	0.42	0.46	0.31	0.33	0.31	0.29	0.37	0.41	0.36	0.38
Qwen [16]	0.33	0.32	0.32	0.36	0.26	0.25	0.26	0.29	0.29	0.34	0.30	0.25	0.29	0.30	0.29	0.30
Similarity [15]	0.21	0.18	0.20	0.19	0.21	0.20	0.21	0.21	0.18	0.17	0.19	0.16	0.20	0.18	0.20	0.19

Table 7

Breakdown of the results according to languages and category: Financial Texts (T), Financial Papers (P), Financial Reports (R)

Language	IT			EN			TR			ALL		
	T	P	R	T	P	R	T	P	R	T	P	R
AMSN [20]	0.89	0.93	0.92	0.88	0.92	0.87	0.88	0.90	0.86	0.88	0.92	0.89
UNITOR [18]	0.88	0.86	0.90	0.88	0.88	0.92	0.87	0.86	0.91	0.88	0.87	0.91
Llama [17]	0.40	0.43	0.27	0.45	0.48	0.37	0.32	0.35	0.25	0.39	0.42	0.29
Qwen [16]	0.33	0.36	0.29	0.24	0.34	0.22	0.31	0.37	0.19	0.29	0.36	0.23
Similarity [15]	0.20	0.19	0.23	0.20	0.22	0.22	0.17	0.22	0.15	0.19	0.21	0.20

AMSN is the best-performing model on both subtasks, with an accuracy of 0.91 and 0.90, as shown in Table 5. UNITOR performs slightly worse, with a score of 0.88 in both tasks. The baselines do not reach a satisfactory accuracy, with the best one, Llama, obtaining a score of 0.37. The Similarity approach yields the worst result, with an accuracy of about 0.20, similar to random choice.

Table 6 shows detailed results. The performance of the participant systems is comparable across languages, and slightly different across difficulty levels. For instance, on Italian data, almost counterintuitively, AMSN performs better on hard questions (0.92) than on easy questions (0.86), whereas UNITOR performs better on the easy questions (0.89) than on the hard ones (0.84). On Turkish data, they both obtain a similar accuracy score across difficulty levels. Similar patterns are also observed in the baselines.

Analyzing the distribution of the answers across the 5 labels, we observe that models from participants do not exhibit any specific bias towards any label and the distribution of their answers follows the distribution of the gold standard.

Table 7 shows results across language and categories. The performance is comparable across systems, however we observe a slight tendency of questions from financial texts (T) to be harder for participants' systems and baselines.

In general, the difference between the results of the participants is small across all subtasks, languages and categories. The size of the underlying models is likely to have a significant impact on the performance, given the gap between approaches that use models above 20B and the baselines, based on models with 3B parameters or less. A remarkable result is that 20B models are competitive against much larger models.

5. Conclusions

The *Prometeia Financial Benchmark* (PFB) shared task was designed to benchmark language models on finance-domain multiple-choice question answering (MCQA) on three languages. We received two full submissions from teams who explored encoder- and decoder-based systems of varying complexity, reaching an accuracy of about 90%. A comparison with the performance obtained by our baselines highlights that PFB is challenging for models smaller than 3B, and that 20B models may be adequate for the task. As future work, we want to investigate more thoroughly the relationship between the models' efficiency and their accuracy. Moreover, we would like to explore the possible benefit of exploiting the difficulty label, which is a feature that the participants did not have, but may be used in ensemble approaches.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI ChatGPT 5.2 in order to: Paraphrase and reword, and Improve writing style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] Y. Dong, F. Wu, K. Zhang, Y. Dai, S. Zhang, W. Ye, S. Chen, Z.-Q. Cheng, Large language model agents in finance: A survey bridging research, practice, and real-world deployment, in: Findings of the Association for Computational Linguistics: EMNLP 2025, 2025, pp. 17889–17907.
- [2] A. T. Khan, S. Li, X. Cao, Bridging finance and AI: a comprehensive survey of large language models in financial system, *Digital Finance* 7 (2025) 679–701.
- [3] H. Kang, X.-Y. Liu, Deficiency of large language models in finance: An empirical examination of hallucination, *arXiv preprint arXiv:2311.15548* (2023). URL: <https://arxiv.org/abs/2311.15548>.
- [4] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, EVALITA 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for

- italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [5] Y. Guo, Z. Xu, Y. Yang, Is ChatGPT a financial expert? evaluating language models on financial natural language processing, arXiv preprint arXiv:2310.12664 (2023). URL: <https://arxiv.org/abs/2310.12664>.
 - [6] P. Winder, C. Hildebrand, J. Hartmann, Biased echoes: Large language models reinforce investment biases and increase portfolio risks of private investors, PloS one 20 (2025) e0325459.
 - [7] R. Jørgensen, O. Brandt, M. Hartmann, X. Dai, C. Igel, D. Elliott, Multifin: A dataset for multilingual financial NLP, in: Findings of the Association for Computational Linguistics: EACL 2023, 2023, pp. 894–909.
 - [8] X. Peng, L. Qian, Y. Wang, R. Xiang, Y. He, Y. Ren, M. Jiang, J. Zhao, H. He, Y. Han, et al., MultiFinBen: A multilingual, multimodal, and difficulty-aware benchmark for financial LLM evaluation, arXiv preprint arXiv:2506.14028 (2025).
 - [9] S. Xue, X. Li, F. Zhou, Q. Dai, Z. Chu, H. Mei, Famma: A benchmark for financial domain multilingual multimodal question answering, arXiv preprint arXiv:2410.04526 (2024).
 - [10] X. Zhang, R. Xiang, C. Yuan, D. Feng, W. Han, A. Lopez-Lira, X.-Y. Liu, M. Qiu, S. Ananiadou, M. Peng, et al., Dólares or dollars? unraveling the bilingual prowess of financial LLMs between Spanish and English, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 6236–6246.
 - [11] Q. Xie, W. Han, Z. Chen, R. Xiang, X. Zhang, Y. He, M. Xiao, D. Li, Y. Dai, D. Feng, et al., Finben: A holistic financial benchmark for large language models, Advances in Neural Information Processing Systems 37 (2024) 95716–95743.
 - [12] G. Matlin, M. Okamoto, H. Pardawala, Y. Yang, S. Chava, Financial language model evaluation (FLaME), in: Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 22633–22679.
 - [13] N. Tatarinov, S. Sukhani, A. Shah, S. Chava, Language modeling for the future of finance: A quantitative survey into metrics, tasks, and data opportunities, arXiv preprint arXiv:2504.07274 (2025).
 - [14] Prometeia, Prometeia financial benchmark: Benchmarking language models in the financial domain, Prometeia Insights, 2025. URL: <https://www.prometeia.com/it/about-us/insights/article/prometeia-financial-benchmark-benchmarking-language-models-in-the-financial-domain-26892170>.
 - [15] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: EMNLP/IJCNLP (1), Association for Computational Linguistics, 2019, pp. 3980–3990.
 - [16] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, J. Tu, J. Zhang, J. Yang, J. Yang, J. Zhou, J. Zhou, J. Lin, K. Dang, K. Bao, K. Yang, L. Yu, L. Deng, M. Li, M. Xue, M. Li, P. Zhang, P. Wang, Q. Zhu, R. Men, R. Gao, S. Liu, S. Luo, T. Li, T. Tang, W. Yin, X. Ren, X. Wang, X. Zhang, X. Ren, Y. Fan, Y. Su, Y. Zhang, Y. Zhang, Y. Wan, Y. Liu, Z. Wang, Z. Cui, Z. Zhang, Z. Zhou, Z. Qiu, Qwen3 technical report, 2025. URL: <https://arxiv.org/abs/2505.09388>. arXiv:2505.09388.
 - [17] A. Grattafiori, et al., The llama 3 herd of models, 2024. URL: <https://arxiv.org/abs/2407.21783>. arXiv:2407.21783.
 - [18] F. Borazio, S. A. Mousavian Anaraki, S. I. Rai, D. Croce, R. Basili, UniTor at PFB: Constrained agentic prompting and self-consistency for financial Multi-Choice QA, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [19] K. Tian, E. Mitchell, H. Yao, C. D. Manning, C. Finn, Fine-tuning language models for factuality, in: ICLR, OpenReview.net, 2024.
 - [20] A. S. Mohammadabad, M. Nazarmohsenifakori, Multilingual economics multiple choice question answering: A comparative study of transformer models, large language models, and hybrid correction systems, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

AMSN at PFB: Overview of the Automated Multilingual Economics Question Answering Task

Ali Saleh Mohammadabad¹, Mehregan Nazarmohsenifakori¹

¹Department of Computer Science and Engineering, University of Bologna, Bologna, Italy

Abstract

This report presents a comprehensive investigation into automated question answering for economics-based multiple choice questions across three languages (English, Italian, and Turkish). We explore three distinct methodological approaches: (1) fine-tuned transformer models, (2) prompt-engineered large language model agents using GPT-4o and GPT-5, and (3) a hybrid mistake detection and correction system. Our results demonstrate that while standalone transformer models achieve approximately 50% accuracy, GPT-4o with chain-of-thought prompting reaches 74% accuracy on English and Italian datasets. A hybrid system combining GPT-4o predictions with transformer-based error correction improves performance to approximately 78%. The newer GPT-5 model achieves superior performance without requiring mistake correction, ultimately serving as our final solution with test set accuracies of 89.11% (English), 91.21% (Italian), and 88.31% (Turkish).

Keywords

Question Answering, Economics, Large Language Models, Chain-of-Thought Prompting, Multilingual NLP, Transformer Models, Error Correction

1. Introduction

The task of automated question answering in specialized domains such as economics presents unique challenges, as it requires not only broad domain knowledge spanning microeconomics, macroeconomics, finance, and econometrics, but also complex multi-step reasoning and robust multilingual understanding. These requirements make economics QA a demanding benchmark for evaluating modern NLP systems.

This report documents our participation in the AMSN shared task at EVALITA 2026, where we systematically explored multiple approaches to address these challenges on a multilingual economics multiple-choice question dataset covering English, Italian, and Turkish. Our investigation progressed through three primary methodologies. We first examined fine-tuned transformer-based models trained directly on the task data, which served as an important baseline for understanding the limitations of supervised fine-tuning with limited domain-specific data. We then evaluated large language models (LLMs) accessed via API, specifically GPT-4o and GPT-5, using carefully designed chain-of-thought prompting strategies that leverage their extensive pre-trained economic knowledge. Finally, we developed a hybrid system that combines LLM predictions with a transformer-based mistake detection and correction module, aiming to capture the complementary strengths of both paradigms.

Through this progression, we demonstrate that advances in base model capability, as exemplified by GPT-5, can outperform both fine-tuned specialised models and hybrid correction approaches, achieving test accuracies of 89.11% (English), 91.21% (Italian), and 88.31% (Turkish).

2. System Description

2.1. Dataset Characteristics

The dataset provided for the AMSN shared task consists of multiple-choice economics questions spanning three languages: English (EN), Italian (IT), and Turkish (TR), each with 1,001 test questions. In addition to the test set, a validation split was provided for system development and hyperparameter

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian

✉ ali.saleh4@studio.unibo.it (A. S. Mohammadabad); mehrega.nazarmohseni@studio.unibo.it (M. Nazarmohsenifakori)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

selection. Each question presents a stem followed by five answer options labelled A through E, where option E typically represents an exclusionary choice such as “None of the above.” The questions cover a wide range of economics topics, including micro- and macroeconomics, finance, and quantitative methods, requiring models to combine domain-specific knowledge with logical reasoning.

2.2. Evaluation Metrics

Performance is measured using classification accuracy:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Questions}} \times 100\%$$

2.3. Approach 1: Transformer Model Training

2.3.1. Architecture and Implementation

We investigated transformer-based models for direct question answering, focusing primarily on multi-lingual architectures suitable for cross-lingual transfer learning.

Models Evaluated

- **mDeBERTa-v3** (Multilingual DeBERTa) [1] – Primary model
- Additional transformer variants for comparison

2.3.2. Results and Limitations

Table 1 presents the validation accuracy achieved by transformer models.

Table 1
Transformer Model Performance

Model	Validation Accuracy
mDeBERTa-v3	~50%

The transformer models demonstrated several limitations:

1. **Complexity of Economic Reasoning:** The models struggled to capture the nuanced reasoning required for economics questions, which often involve multi-step logical deduction and domain-specific knowledge.
2. **Limited Training Data:** The available training data was insufficient for the model to learn robust patterns for all question types and economic concepts.
3. **Domain Knowledge Gap:** Fine-tuning alone proved inadequate for encoding the breadth of economic knowledge required.

2.4. Approach 2: Large Language Model Agents with Prompt Engineering

2.4.1. Motivation

Given the limitations of fine-tuned transformer models, we explored leveraging the pre-trained knowledge embedded in large language models through carefully designed prompting strategies. We evaluated two models accessed via the OpenAI API: GPT-4o [2] (snapshot gpt-4o-2024-08-06) and the newer GPT-5 (exact snapshot version to be specified by the authors; see Section 2.4.3).

2.4.2. Prompting Strategies

We systematically evaluated five distinct prompting strategies on a sample of 50 validation questions. Table 2 summarizes the results.

Table 2
Prompting Strategy Comparison (50 validation questions)

Strategy	Score	Accuracy	Rank
Chain of Thought	45/50	90.0%	Best
Expert Reasoning	45/50	90.0%	Best
Careful Analysis	44/50	88.0%	
Baseline	43/50	86.0%	
Step-by-Step	43/50	86.0%	

2.4.3. Optimal Prompt Design

The chain-of-thought prompting strategy [3] emerged as the top performer. The complete prompt implementation consists of a system prompt that establishes the model’s expertise and reasoning framework, combined with a user prompt containing the formatted question.

System Prompt. The system prompt establishes the model’s role and reasoning methodology:

You are an expert economist with deep knowledge in microeconomics, macroeconomics, finance, and econometrics.

For this multiple choice question:

1. First, carefully analyze each statement or option
2. Apply your economic knowledge to evaluate correctness
3. Consider edge cases and common misconceptions
4. After your analysis, provide your final answer

IMPORTANT: End your response with "ANSWER: X" where X is the letter (A, B, C, D, or E).

Question Formatting. Each question is formatted as a user prompt with the question text followed by lettered options (A through E).

API Configuration. The prompts were executed with temperature 0.1 for consistent responses and a maximum of 1500 output tokens for detailed reasoning. We used the OpenAI Chat Completions API with model identifiers gpt-4o-2024-08-06 (GPT-4o) and gpt-5-YYYY-MM-DD (GPT-5; replace with the exact snapshot version used).¹

Answer Extraction. Responses were parsed using multiple extraction patterns including "ANSWER: X", "The correct answer is X", and fallback to the last letter A-E in the response, achieving near-perfect parsing accuracy.

2.4.4. Full Dataset Evaluation

Applying the chain-of-thought strategy to the complete dataset across all three languages yielded the results shown in Table 3.

2.4.5. Key Observations

Several important patterns emerged from the full-dataset evaluation. First, the chain-of-thought approach does more than improve accuracy: it exposes the model’s step-by-step reasoning, providing interpretability and enabling downstream error analysis by inspecting the rationale behind each prediction. Second, the same prompting strategy generalised effectively across all three languages, achieving

¹Readers wishing to reproduce results should note that OpenAI API model behaviour may vary across snapshot versions. The exact version strings used in our experiments should be substituted here by the authors.

Table 3
LLM Performance Across Languages (Validation Set)

Model	English (EN)	Italian (IT)	Turkish (TR)
GPT-4o	74%	74%	65%
GPT-5	82%	81%	73%

consistent performance on English and Italian. Third, a Turkish performance gap is evident, with accuracies 9–10 percentage points lower than English and Italian for both models, suggesting that coverage of Turkish economics terminology and concepts is comparatively weaker in the models’ pre-training data. Finally, GPT-5 demonstrates substantial generational improvements over GPT-4o, with gains of approximately 8 percentage points across all three languages, underscoring the impact of advances in base model capability on downstream task performance.

2.5. Approach 3: Hybrid Mistake Detection and Correction

2.5.1. Rationale

While GPT-4o achieved respectable 70-75% accuracy, we investigated whether a specialized model trained to detect and correct GPT-4o’s mistakes could push performance beyond this baseline.

2.5.2. System Architecture

The hybrid system, implemented in `Finance_Concepts_Transformer_Updated (1).ipynb`, consists of two components:

1. **Mistake Detector:** A transformer model trained to identify when the LLM’s prediction is likely incorrect
2. **Answer Corrector:** A model that predicts the correct answer when mistakes are detected

The system uses the following input features:

- Question text
- All answer options
- GPT-4o’s prediction
- **GPT-4o’s reasoning** (extracted from chain-of-thought output)

2.5.3. Training Data

The hybrid components were trained using GPT-4o predictions on the validation set, for which ground truth labels were available. For the mistake detector, binary labels were derived by comparing GPT-4o’s predicted answer against the ground truth: predictions matching the correct answer were labelled as correct, while mismatches were labelled as mistakes. For the answer corrector, the ground truth answer label for each question was used as the target class in a five-way multi-class classification setting.

2.5.4. Results

The hybrid system applied to GPT-4o predictions achieved approximately **77-79% accuracy**, representing a 3-5 percentage point improvement over standalone GPT-4o predictions on English and Italian datasets.

We also evaluated whether the mistake detection system could improve GPT-5 predictions. However, the correction mechanism did not yield improvements, suggesting that GPT-5’s reasoning capabilities already capture the patterns the mistake detector was designed to identify.

3. Results

3.1. Validation Performance

Table 4 presents a comprehensive comparison of all approaches on the validation set.

Table 4
Performance Comparison Across Approaches (Validation Set)

Approach	EN	IT	TR	Avg
Transformer (mDeBERTa-v3)	50%	50%	50%	50.0%
GPT-4o (Chain-of-Thought)	74%	74%	65%	71.0%
GPT-4o + Mistake Correction	78%	77%	—	77.5%
GPT-5 (Chain-of-Thought)	82%	81%	73%	78.7%
GPT-5 + Correction (no improvement)	82%	81%	73%	78.7%

3.2. Final Test Set Results

After finalizing our approach with GPT-5, we evaluated the model on the held-out test set of 1,001 questions per language. Table 5 presents the final test accuracies.

Table 5
Final Test Set Performance (1,001 questions per language)

Language	Correct	Incorrect	Accuracy	Improvement
English (EN)	892	109	89.11%	+7.11 pp
Italian (IT)	913	88	91.21%	+10.21 pp
Turkish (TR)	884	117	88.31%	+15.31 pp
Average	896.3	104.7	89.54%	+10.84 pp

The test set results demonstrate exceptional performance, with GPT-5 achieving nearly 90% average accuracy across all three languages. Notably, the test performance significantly exceeds validation performance, with improvements ranging from 7.11 to 15.31 percentage points.

3.3. Answer Distribution Analysis

Table 6 presents the answer distribution across all three languages.

Table 6
Predicted Answer Distribution by Language (1,001 questions each)

Language	A	B	C	D	E	Total
English	26.9%	27.6%	28.1%	16.2%	1.3%	100%
Italian	27.5%	28.2%	27.1%	14.9%	2.4%	100%
Turkish	27.3%	26.2%	29.7%	15.8%	1.1%	100%
Average	27.2%	27.3%	28.3%	15.6%	1.6%	100%

Key observations from the distribution analysis:

- Options A, B, and C account for approximately 82-83% of all predictions across languages
- Option D represents only 15-16% of predictions
- Option E (“None of the above”) appears in only 1-2% of predictions
- The remarkable similarity in distribution patterns across languages indicates robust multilingual performance

4. Discussion

4.1. Comparative Analysis

Our investigation reveals a clear performance hierarchy:

1. **GPT-5 Test Performance (89.54% avg):** Achieves exceptional performance on the held-out test set, significantly exceeding validation performance (78.7%).
2. **GPT-5 Validation (78.7% avg):** Strong performance during development, selected as final solution.
3. **GPT-4o + Mistake Correction (77.5% avg):** Demonstrates that targeted error correction can improve baseline LLM performance.
4. **GPT-4o with Chain-of-Thought (71.0% avg):** Leverages pre-trained knowledge effectively, outperforming fine-tuned models by 21 percentage points.
5. **Fine-tuned Transformers (50%):** Limited by training data and domain knowledge constraints.

4.2. Key Success Factors

The results across our three approaches point to several factors that drove performance. The most decisive factor was the breadth of pre-trained knowledge embedded in large language models: GPT-4o and GPT-5 have been exposed to extensive economics literature during pre-training, providing a foundation for domain reasoning that supervised fine-tuning of smaller models on limited task data simply cannot replicate.

Chain-of-thought prompting contributed not only to accuracy but also to reasoning transparency, making the system’s decision process interpretable and facilitating downstream error analysis. For the GPT-4o baseline specifically, the hybrid correction system proved beneficial by exploiting complementary strengths: the LLM contributes broad knowledge and reasoning, while the specialist transformer contributes targeted pattern recognition for detecting and correcting systematic errors. Finally, the results illustrate the impact of model generation: GPT-5’s improvements over GPT-4o were large enough to make the separate correction mechanism redundant, suggesting that architectural and training advances in frontier models can subsume the role of specialised post-processing.

4.3. Analysis of Results

The results reveal several noteworthy patterns. For GPT-4o, the mistake detector successfully identifies systematic errors in the model’s predictions, and the subsequent correction step yields measurable accuracy gains of 3–5 percentage points on English and Italian. This confirms that GPT-4o makes identifiable, recurring mistakes that a specialised classifier can learn to detect.

In contrast, the same correction mechanism applied to GPT-5 predictions produced no improvement. This outcome indicates that GPT-5’s stronger reasoning capabilities already address the failure modes that the mistake detector was designed to capture, making the additional correction layer redundant. Consequently, GPT-5 with chain-of-thought prompting was selected as the final submission system, without any post-processing.

The most striking pattern in the results is the gap between validation and test performance, discussed in detail in Section 4.4. Rather than reflecting overfitting, this improvement is consistent across all three languages and all metrics, suggesting genuine differences in question characteristics between the two splits. Given GPT-5’s superior standalone performance, it was the natural choice for the final submission.

4.4. Validation-to-Test Performance Gap

A notable feature of our results is the consistent improvement from validation to test performance, ranging from +7.11 percentage points (English) to +15.31 percentage points (Turkish). While the

magnitude of this gain is striking, we do not believe it reflects data leakage or overfitting, for the following reasons.

First, GPT-5 was not fine-tuned on the validation set in any way; the only information derived from validation data was the selection of the chain-of-thought prompting strategy, which was evaluated on a small 50-question sample. The model itself was called via API in a zero-shot inference mode on both splits. Second, the improvement is consistent across all three languages, including Turkish, which argues against a language-specific anomaly.

The most plausible explanation is a distributional difference between the validation and test splits. In particular, the test questions may draw more heavily on canonical economic theory and textbook-level definitions that are well represented in GPT-5's pre-training corpus, whereas the validation questions may include a higher proportion of edge cases, ambiguous phrasing, or niche topics. A secondary contributing factor could be question phrasing: test questions may be more unambiguously worded, making chain-of-thought reasoning more reliable.

We acknowledge that a full investigation of this gap would require access to ground-truth labels for the validation set alongside a systematic analysis of question-level difficulty, topic distribution, and linguistic complexity across the two splits. We leave this as an important direction for future work, and encourage the task organisers to share validation gold labels to enable the research community to conduct such analyses.

5. Conclusion

This work demonstrates a systematic exploration of approaches to multilingual economics question answering, progressing from fine-tuned transformers (50%) to GPT-4o (71%) to GPT-4o with mistake correction (77.5%) to GPT-5 validation (78.7%), and ultimately achieving exceptional test performance (89.54% average).

Three key findings emerge from this work. First, advances in base model capability can subsume the benefits of specialised post-processing: GPT-5 outperformed GPT-4o with hybrid correction, demonstrating that investing in stronger foundation models can be more productive than engineering correction layers on top of weaker ones. Second, transformer-based mistake correction is effective but model-dependent: the correction system improved GPT-4o by approximately 6.5 percentage points on average but yielded no gain for GPT-5, confirming that such systems are most valuable when the underlying model has identifiable, systematic weaknesses. Third, GPT-5 with chain-of-thought prompting achieved strong test accuracies of 89.11% on English, 91.21% on Italian, and 88.31% on Turkish (average 89.54%).

The exceptional cross-lingual performance (88-91%) validates the robustness of our GPT-5 approach across all three languages. Italian achieved the highest accuracy (91.21%), while Turkish showed the most significant improvement from validation to test (+15.31 pp), effectively closing the performance gap.

Future work should focus on understanding the validation-to-test performance gap, investigating whether similar performance can be achieved with cost-effective open-source alternatives, and exploring the remaining 10% error cases to identify systematic failure patterns.

6. Declaration on Generative AI

In the preparation of this work, the authors utilized the following generative AI tools:

Research Subject (Models Under Investigation):

- **GPT-4o and GPT-5** (OpenAI's large language models) were the primary subject of investigation for multilingual economics question answering. These models were evaluated through their APIs to generate predictions on economics multiple-choice questions across English, Italian, and Turkish languages using chain-of-thought prompting strategies.

- **mDeBERTa-v3** transformer model was employed for baseline comparisons and as part of the hybrid mistake detection and correction system implemented in `Finance_Concepts_Transformer_Updated (1).ipynb`.

Writing Assistance:

- **Claude (Anthropic)** was used to assist in drafting portions of this manuscript, including structuring sections, formatting LaTeX code, refining language clarity, and organizing the presentation of results and methodologies. The authors reviewed, edited, and verified all AI-generated content for accuracy and appropriateness.

The experimental design, hypothesis formulation, data analysis, interpretation of results, and scientific conclusions were conducted entirely by the human authors. All code implementation, data processing, model evaluation, and statistical analysis were performed by the authors without AI assistance.

Acknowledgments

We thank the organizers of the FinanceNLP shared task for providing the multilingual economics question answering dataset and evaluation framework.

References

- [1] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, in: International Conference on Learning Representations (ICLR), 2023.
- [2] OpenAI, Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).
- [3] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, D. Zhou, Chain-of-thought prompting elicits reasoning in large language models, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 24824–24837.

UniTor at PFB: Constrained Agentic Prompting and Self-Consistency for Financial Multi-Choice QA

Federico Borazio¹, Seyed Alireza Mousavian Anaraki¹, Shahid Iqbal Rai¹, Danilo Croce¹ and Roberto Basili¹

¹Department of Enterprise Engineering, University of Rome Tor Vergata
Via del Politecnico 1, 00133, Rome, Italy

Abstract

General-purpose Large Language Models (LLMs) often struggle to combine domain specificity with robust cross-lingual generalization, particularly in high-stakes domains such as finance. In this paper, we present our contribution to the PFB: Prometeia Financial Benchmark, tackling Financial Multi-Choice Question Answering (MCQA) in English, Italian, and Turkish. We introduce a constrained agentic workflow that improves reliability exclusively through inference-time control, without fine-tuning or external tools. Our system implements a hierarchical pipeline based on semantic routing to specialized reasoning archetypes (Analyst, Fact-Checker, Researcher), parallel stochastic sampling to expose reasoning instability, and a deterministic algorithmic consensus mechanism that promotes answers stable under resampling while suppressing hallucinations. Extensive evaluation on the blind test set shows that this architecture consistently closes the cross-lingual performance gap, enabling a mid-sized open model (GPT-OSS-20B) to reach state-of-the-art accuracy ($\approx 88\%$) across all three languages. Remarkably, our approach matches or slightly outperforms much larger proprietary models, while remaining robust in low-resource settings such as Turkish. Ablation results demonstrate that stochastic self-consistency is essential for handling structural complexity, particularly tabular and boolean reasoning, and that controlled aggregation provides a scalable, training-free alternative to model scaling and domain-specific fine-tuning.

Keywords

Large Language Models, Agentic Workflows, Ensemble Inference, Financial NLP

1. Introduction

In recent years, Natural Language Processing has witnessed an unprecedented paradigm shift driven by the advent of Large Language Models (LLMs). While general-purpose models have demonstrated impressive performance across a wide range of tasks, from logical reasoning [1] to creative generation [2], their effectiveness in highly specialized domains, such as finance, remains an open question [3, 4, 5]. The financial domain presents unique challenges that go beyond standard linguistic comprehension. It requires the ability to interpret complex technical terminology, grasp subtle regulatory nuances, and perform rigorous numerical reasoning over structured data [6]. While recent literature has begun to explore the adaptation of LLMs to financial language, the vast majority of studies focus on English, leaving other languages, such as Italian, significantly underrepresented. This creates a critical gap: although multilingual models promise universality, it is unclear whether they can compete with specialized baselines [7] in vertical sectors where local market specificities play a crucial role.

In this context, the present work outlines our contribution to the **PFB: Prometeia Financial Benchmark**¹, a shared task organized within EVALITA 2026 [8] specifically focused on Financial Multi-Choice Question Answering (MCQA). This benchmark serves as a stress test for model capabilities in a constrained, real-world scenario, offering a high-quality multilingual dataset (Italian, English,

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ borazio@ing.uniroma2.it (F. Borazio); seyedalireza.mousaviananaraki@students.uniroma2.eu (S. A. Mousavian Anaraki); rjshahidrai@gmail.com (S. I. Rai); croce@info.uniroma2.it (D. Croce); basili@info.uniroma2.it (R. Basili)

🆔 0009-0000-0193-2131 (F. Borazio); 0009-0007-1044-9978 (S. A. Mousavian Anaraki); 0009-0006-1868-8407 (S. I. Rai); 0000-0001-9111-1950 (D. Croce); 0000-0001-5140-0694 (R. Basili)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://www.prometeia.com/it/about-us/insights/article/prometeia-financial-benchmark-benchmarking-language-model-s-in-the-financial-domain-26892170>

Turkish) annotated by complexity. The task requires systems to select the correct option among a fixed set of candidates for each question. Each instance consists of a natural language question, five answer options (A–E), and a single correct label, and is evaluated using accuracy-based metrics stratified by difficulty level. Unlike generic QA datasets, the PFB task challenges models with heterogeneous reasoning requirements. Financial questions exhibit strong structural heterogeneity, ranging from conceptual queries requiring domain knowledge, to boolean verification tasks and complex tabular reasoning scenarios demanding numerical extraction and calculation.

The design choices of the proposed architecture are motivated by two empirical observations about LLM behavior in specialized MCQA tasks. First, financial questions exhibit strong structural heterogeneity, and no single prompting strategy is uniformly effective across conceptual, boolean, and tabular reasoning. Second, LLM errors in such settings are often *unstable*: hallucinated or incorrect answers tend to vary across stochastic generations, whereas correct solutions are more likely to reappear consistently. Based on this observation and inspired by [9], our approach treats stochastic sampling not as a source of noise, but as a mechanism to probe the stability of candidate answers. By generating multiple reasoning variants for the same query and aggregating them deterministically, the system implicitly favors answers that are robust under resampling, while suppressing spurious or idiosyncratic generations. The iterative refinement step further enforces this principle by constraining the search space when consensus is weak, effectively amplifying stable reasoning patterns and mitigating hallucinations.

We evaluate this architecture across a spectrum of models, from lightweight open-source solutions (e.g., LLaMA 3.1 8B) to large-scale proprietary engines (e.g., DeepSeek v3.1). This comparative analysis allows us to investigate the trade-offs between model size, inference efficiency, and cross-lingual transfer in the financial domain.

Our study is driven by the following core research questions:

- **RQ1:** Can general-purpose LLMs achieve domain-competent performance in specialized financial tasks relying solely on structured reasoning across LLM ensembles?
- **RQ2:** How does the structural complexity of the question (e.g., tabular vs. conceptual, easy vs. hard) impact performance across different languages?
- **RQ3:** Can reasoning patterns and financial knowledge effectively favor the transfer from an English-centric pre-training to the target Italian and Turkish contexts?
- **RQ4:** Do stochastic sampling and hybrid aggregation strategies significantly improve reliability compared to standard greedy decoding?

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature on LLMs for reasoning, financial domain adaptation, and agentic workflows. Section 3 details our proposed system architecture, describing the semantic routing mechanism, the specialized reasoning modules, and the aggregation strategy. Section 4 presents the experimental setup and discusses the results, providing a granular analysis of model performance across languages and complexity levels. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Related Work

Large language models (LLMs) have achieved significant advances in QA across multiple domains [10, 11]. In the educational context, elimination-based reasoning with LLMs has been applied to MCQA [12], incorporating structured prompting and intermediate decision steps with chain-of-thought (CoT) reasoning to eliminate incorrect options sequentially. In the medical domain, approaches such as reasoning with LLMs [13] and collaboration among multiple LLMs [14] have been explored to enhance accuracy and consistency. These efforts include ensemble reasoning strategies that refine intermediate steps and reduce inconsistencies, demonstrating the potential of LLMs to perform complex, multi-step reasoning in specialized question-answering tasks.

Despite these advances, increasing the size of the model alone does not substantially improve performance in complex reasoning tasks. CoT prompting has therefore been proposed, guiding the

model through intermediate reasoning steps to solve tasks sequentially. Several enhancements have further improved CoT performance [15]. A simple yet effective method is self-consistency [16], which generates multiple reasoning paths and determines the final answer via majority voting [17]. Additional approaches include calibrating reasoning paths with a verifier [18] and using dynamic voting strategies for more efficient reasoning in LLMs [15].

In financial MCQA, the type of questions and the format of the data are particularly important. Tables and structured data are central to many financial tasks, such as analyzing balance sheets or performance reports. While traditional LLMs excel at text-based reasoning, they struggle to interpret data presented in image or table formats. Recent multimodal LLMs have begun to bridge this gap by processing both text and images; however, their performance on table images remains limited and lacks domain specificity. To address this, [19] introduces FinTab-LLaVA, a finance-specific multimodal LLM designed for table understanding using FinTMD.

Intelligent AI agents are becoming a core component of autonomous, goal-oriented systems that can perceive their environment, reason over information, learn from data, and execute actions. In the financial domain in particular, artificial intelligence is fundamentally transforming financial infrastructures and services. Agent-based AI approaches are now widely applied in areas such as algorithmic trading, fraud and anomaly detection, credit scoring and risk evaluation, automated investment advisory services, and regulatory technology (RegTech) for compliance and monitoring [20].

The advent of LLMs has further driven the development of intelligent agents, where the LLM functions as the primary reasoning component or “brain”. These models offer strong capabilities in natural language understanding, text generation, reasoning, and knowledge retrieval. Agent architectures typically integrate LLMs with external tools, memory components, and additional processing modules [21]. Such LLM-based agents have also been applied to financial information retrieval and to the augmentation of knowledge-intensive work in finance [22, 23].

While agent capabilities provide an important foundation, the architectural organization of agents is equally critical. In the literature, agent-based systems are commonly structured as multi-agent systems (MAS), hierarchical agents, or hybrid architectures. MAS comprises multiple autonomous agents interacting within a shared environment, supporting collaborative or competitive problem solving through coordination and distributed decision making [24]. Hierarchical architectures address long-horizon tasks by decomposing them into multi-level goals [25], while hybrid systems integrate diverse AI techniques, such as LLM reasoning, retrieval, external tools, and specialized models, within a unified framework to leverage complementary strengths [26]. In the context of financial QA, FinAgentBench introduces the first large-scale benchmark for evaluating retrieval with multi-step reasoning in finance, a setting referred to as agentic retrieval [27].

While these approaches demonstrate the effectiveness of structured prompting, self-consistency, and agent-based reasoning, most prior work focuses either on open-ended QA, retrieval-augmented settings, or domain-specific fine-tuning. In contrast, the Financial MCQA setting considered in this work imposes a strict closed-set decision constraint, where systems must select a single option without access to external tools or additional supervision. Moreover, existing agentic frameworks typically emphasize autonomy, tool usage, or long-horizon planning, whereas our approach adopts a deliberately constrained agentic design tailored to inference-time robustness. By combining task-aware routing, stochastic reasoning, and deterministic aggregation, our work bridges ensemble-based self-consistency methods and agentic workflows in a unified architecture specifically designed to mitigate hallucinations in multilingual financial MCQA.

3. Description of the System

Financial MCQA exposes two recurrent limitations of LLMs. First, financial questions exhibit strong structural heterogeneity: conceptual queries, boolean verification tasks, and tabular reasoning problems require fundamentally different reasoning strategies. Second, model predictions in this setting are often unstable: early reasoning errors or hallucinations in an early inference pass can disproportionately

affect the final answer. As a consequence, a monolithic prompting strategy and single-pass inference are insufficient to ensure robust performance.

To address these issues, we propose a modular inference architecture that decomposes reasoning into specialized stages and consolidates multiple stochastic hypotheses into a single, stable decision. We refer to this design as a **constrained agentic workflow**. The term *agentic* reflects the use of role-based specific stages and explicit orchestration of reasoning behaviors, while *constrained* emphasizes that the system deliberately avoids autonomous tool usage, open-ended planning, memory accumulation, or unbounded interaction loops. All components operate within a fixed and fully controlled inference graph, ensuring reproducibility, stability, and low latency.

Within this framework, each input question is first analyzed and routed to a question-specific reasoning role. The selected role is then instantiated multiple times under stochastic sampling to generate diverse reasoning hypotheses. Rather than treating redundancy as a source of noise, the system exploits it to probe the stochastic stability of candidate answers: correct solutions tend to recur across independent generations, while hallucinated or brittle answers are less consistent. A deterministic aggregation stage subsequently consolidates these hypotheses, favoring answers that are robust under resampling and suppressing spurious outputs.

3.1. Overall Architecture

The proposed system follows a three-stage orchestration pipeline, illustrated in Figure 1, composed of *semantic routing*, *specialized reasoning*, and *hybrid decision aggregation*.

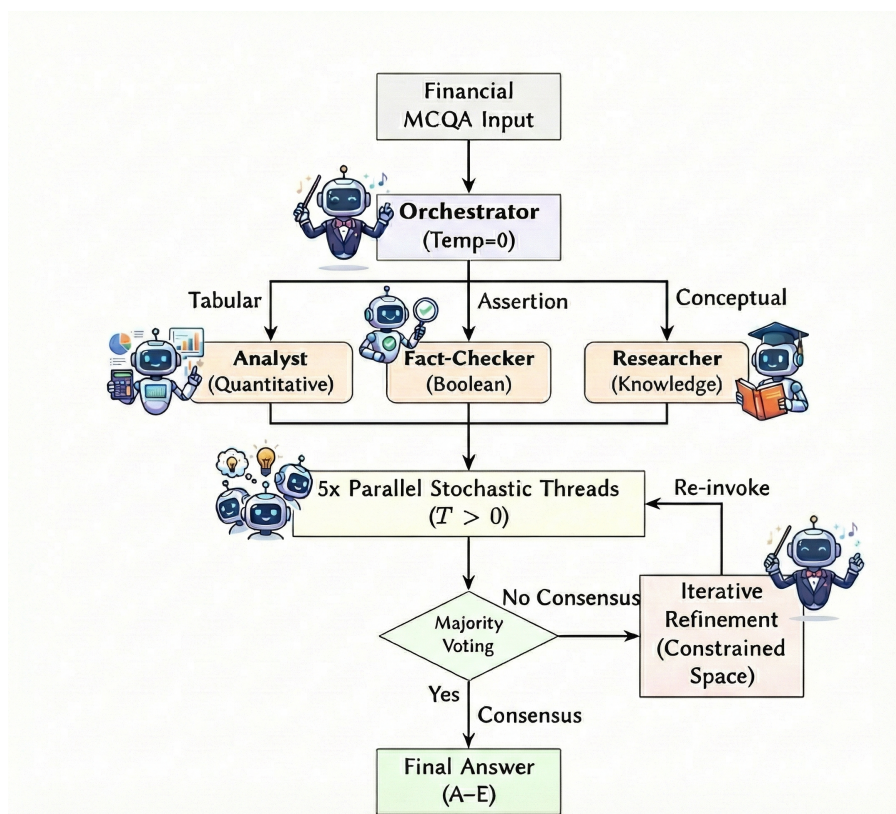


Figure 1: Architectural overview of the proposed constrained agentic workflow. Inference is decomposed into a deterministic routing stage, followed by parallel stochastic reasoning and a controlled aggregation mechanism. The Orchestrator assigns each query to a specialized reasoning archetype. Multiple stochastic generations probe the stability of candidate answers, while hybrid aggregation consolidates robust solutions and suppresses unstable or hallucinated outputs.

The pipeline begins with the **Semantic Routing** stage, implemented by a lightweight LLM-based *Orchestrator*. Its role is to prevent a one-size-fits-all reasoning strategy from being applied to structurally

different questions. Given an input query, the Orchestrator deterministically classifies it into one of three reasoning archetypes: conceptual (*Researcher*), boolean verification (*Fact-Checker*), or tabular reasoning (*Analyst*). Determinism at this stage is critical, as routing errors would systematically propagate to all subsequent reasoning threads.

Following routing, the system enters the **Specialized Reasoning** phase. Instead of producing a single solution, the selected reasoning role is instantiated in multiple parallel inference threads. Each thread operates with non-zero temperature ($T = 0.7$), inducing variability in intermediate reasoning steps while preserving the same task-specific prompt. This design simulates a panel of independent experts analyzing the same problem and enables the system to explore alternative reasoning paths.

Finally, the **Hybrid Aggregation** stage consolidates the outputs of the parallel reasoning threads into a single prediction. Majority voting is used to select the most stable answer, effectively filtering out isolated reasoning errors. When consensus is weak, an iterative refinement mechanism is triggered to constrain the answer space and force convergence. Through this process, stochastic exploration is progressively narrowed into a deterministic decision, improving robustness without introducing open-ended inference loops.

In the remainder of this section, we describe each stage of the workflow in detail, clarifying its role, design rationale, and interaction with the other components.

Routing and Task Decomposition. The first stage of the pipeline performs *semantic routing*, whose purpose is to associate each input question with a reasoning strategy that is appropriate to its dominant structure. The routing scheme is not derived from an external taxonomy, but is introduced by the authors as a pragmatic abstraction based on an empirical inspection of the PFB development set. Through this analysis, we observed that the vast majority of questions could be naturally grouped according to their primary reasoning requirement. We therefore define three *reasoning archetypes*: (i) *Analyst*, for questions involving numerical reasoning or explicit value extraction from tables or structured financial statements; (ii) *Fact-Checker*, for questions presenting multiple assertions that must be individually validated as true or false; and (iii) *Researcher*, for conceptual or theoretical questions primarily relying on domain knowledge. These archetypes are not intended as exhaustive or universally valid categories, but as a lightweight operational decomposition tailored to the structure of the benchmark. Routing is implemented by a lightweight LLM-based *Orchestrator* operating deterministically ($T = 0$), ensuring stable and reproducible classification decisions. On the PFB development set, this heuristic routing achieved near-perfect agreement with the manually inspected assignment, reflecting the strong regularity of question structures in the benchmark rather than an intrinsic generalization guarantee. To mitigate potential misclassification in less regular or unseen cases, a conservative fallback strategy is employed. When the Orchestrator fails to identify a clear dominant archetype, the system defaults to a generic inference configuration using a neutral prompt that does not enforce any role-specific reasoning constraints. This routing component is implemented through a dedicated prompt template; the full prompt specifications are reported in Appendix A.1.

Specialized Reasoning Modules. After routing, the selected reasoning archetype is instantiated multiple times to generate a set of independent hypotheses. Each archetype (*Analyst*, *Fact-Checker*, *Researcher*) is implemented through a *distinct, role-specific prompt* that encodes a different reasoning strategy (quantitative extraction, boolean verification, or conceptual synthesis, respectively). Within a given archetype, all parallel instances share the same prompt and inference configuration, but operate under stochastic sampling ($T = 0.7$). As a result, the system produces multiple alternative reasoning traces corresponding to the *same reasoning strategy*, rather than mixing heterogeneous behaviors. This design can be interpreted as a panel of independent virtual experts operating under the same methodological guidelines (i.e., the specific prompt constraints), each providing a separate judgment on the same question. Correct answers tend to emerge consistently across repeated samples, while hallucinated or fragile conclusions are less stable. By explicitly repeating the same role-conditioned reasoning process, the system enables downstream aggregation to assess answer robustness under

resampling, rather than relying on a single reasoning trace. All reasoning modules terminate generation via EOS detection and are constrained to output a single option label (A–E), ensuring consistent parsing and reliable aggregation. The full prompt templates defining each reasoning archetype are reported in Appendix A.1.

Aggregation Policies. The final decision is obtained through a deterministic aggregation of the $N = 5$ stochastic reasoning outputs produced for each question. Each reasoning instance returns a single option label (A–E), and these labels are interpreted as independent *votes*. We define *consensus* as the emergence of a dominant option that is selected by a strict majority of the instances. Accordingly, the system applies **Majority Voting**: the option receiving the highest number of votes is selected as the final answer. This mechanism operationalizes self-consistency by favoring answers that remain stable across resampled reasoning paths, while suppressing isolated hallucinations that occur sporadically in individual generations. A special handling is introduced for the option “None of the above” (Option E). Empirical inspection revealed a systematic *negative refusal bias*, whereby models tend to select E as a fallback under uncertainty. To prevent spurious refusals, Option E is accepted as a valid consensus *only* if it is unanimously supported (5 out of 5 votes). If this condition is not met, votes for E are discarded, and the majority decision is recomputed over the remaining options. In cases where majority voting does not yield a clear consensus, specifically, when the vote margin between the two most frequent options is ≤ 1 or a perfect tie occurs, the system activates a deterministic **Iterative Refinement** procedure. During refinement, the reasoning modules are re-invoked with a constrained version of the original prompt, in which low-support options are explicitly masked (e.g., replaced with “do not consider this option”). This forces the model to re-evaluate the question within a reduced answer space, amplifying stable reasoning patterns. The refinement loop is bounded to a maximum of two additional rounds to guarantee termination. If no consensus is reached after the final iteration, the system deterministically selects the option with the highest cumulative vote count, breaking residual ties via lexicographical order.

System Details. In line with the objectives of the shared task, which aims to assess the behavior of both large and small language models in the financial domain, we evaluate a deliberately diverse set of LLMs differing in scale, openness, and architectural design. This selection is intended to analyze how model capacity and accessibility interact with our inference-time architecture, rather than to optimize performance through model-specific tuning. Specifically, the evaluated models allow us to compare reasoning robustness, cross-lingual transfer, and inference efficiency under a unified and controlled experimental setup.

LLaMA 3.1 8B [28] is included as a lightweight open-source baseline, reflecting the shared task’s emphasis on evaluating compact models alongside larger ones.² Its limited parameter budget provides a lower-bound reference for assessing how much of the observed performance gains can be attributed to architectural orchestration rather than raw model scale. GPT-OSS-20B [29] serves as the primary open-source reference model. It is a reasoning-oriented Mixture-of-Experts architecture with 21B total parameters (approximately 3.6B active per inference), striking a balance between expressive capacity and computational efficiency.³ This model is particularly well-suited to our setting, as it enables extensive stochastic sampling and parallel inference without prohibitive latency.⁴ DeepSeek v3.1 [30] is included as a large-scale proprietary upper bound, featuring a Mixture-of-Experts architecture with 671B total parameters and approximately 37B active per inference.⁵ Its strong performance on complex reasoning tasks makes it a suitable reference to assess whether inference-time orchestration can narrow the gap between mid-sized open models and state-of-the-art proprietary systems. Inference for DeepSeek is

²<https://huggingface.co/meta-llama/Llama-3.1-8B>

³<https://huggingface.co/openai/gpt-oss-20b>

⁴In our setup, the `reasoning_effort` parameter is set to `low` for the Orchestrator and `medium` for the specialized reasoning modules.

⁵<https://www.deepseek.com>

performed via the Ollama Cloud managed service, enabling access to datacenter-scale hardware without local resource constraints.⁶

All models are evaluated *as-is*, without any additional fine-tuning or domain adaptation. Consequently, all performance differences observed in the experiments are attributable solely to the proposed system architecture, prompt specialization, and deterministic aggregation strategies. To ensure comparability across languages and isolate cross-lingual reasoning effects, the same English prompts are used for English, Italian, and Turkish inputs. This design choice reflects the fact that most large-scale pre-training and reasoning alignment occur in English, and allows us to explicitly measure how well structured reasoning patterns transfer to lower-resource languages. Overall, this experimental setup enables a controlled investigation of how model scale and openness interact with constrained agentic prompting and self-consistency in multilingual financial MCQA. The orchestration logic is implemented via a lightweight custom controller, without relying on external agentic frameworks, in order to retain full control over inference flow and reproducibility.

4. Results & Discussion

This section presents a comprehensive evaluation of the proposed framework across English, Italian, and Turkish. The analysis is structured to address the research questions introduced in Section 1, covering architectural effectiveness and self-consistency (RQ1, RQ4), cross-lingual transfer (RQ3), and sensitivity to question structure and difficulty (RQ2).

4.1. Experimental Setup

We adopt a strict separation between system calibration and final evaluation, using only the official PFB datasets. All architectural choices, prompt designs, and routing heuristics are defined on the Development Set, while all reported results are obtained on the blind Test Set.

Data. The Development Set consists of 500 instances per language. For this subset, question archetypes (*Researcher*, *Fact-Checker*, *Analyst*) are manually identified and used exclusively to validate the routing strategy. Question difficulty is *not* used during system design or calibration. Difficulty labels are employed only *a posteriori* for stratified result analysis. The development data include 177 conceptual, 148 assertion-based, and 175 tabular questions. Final evaluation is conducted on the official blind Test Set, comprising 1,001 unlabelled instances per language. The test set preserves the structural heterogeneity of the task, with 303 conceptual, 371 assertion-based, and 327 tabular questions. Difficulty annotations (166 easy, 771 medium, 64 hard) are used exclusively for reporting and analysis, and do not influence inference or aggregation decisions.

Evaluation Protocol and Baselines. We evaluate the proposed framework through a multi-faceted analysis that considers overall accuracy, cross-lingual transfer, and robustness with respect to question structure and difficulty. All results are reported on the blind test set and are stratified by language, difficulty level, and question archetype when relevant. To isolate the contribution of each architectural component, we compare the full system against a set of progressively enriched baselines. These configurations allow us to disentangle the effects of routing, modular prompting, stochastic sampling, and aggregation, and to quantify the benefits of each design choice.

We consider the following inference configurations:

- **Monolithic Zero-Shot (Baseline).** A single-pass inference using a generic instruction-following prompt, without explicit reasoning steps or task decomposition. This setting represents the lower bound and measures the model’s raw financial QA capability.

⁶<https://ollama.com/cloud>

- **Integrated Chain-of-Thought (Single-Pass).** A stronger baseline in which the model is prompted to internally identify the question type and produce a step-by-step solution within a single inference pass. This configuration evaluates whether in-context reasoning alone can replace explicit architectural modularization.
- **Orchestrated Greedy Inference (No Voting).** The full routing and role-specific prompting architecture is activated, but inference is performed deterministically ($N = 1, T = 0$). This setting isolates the impact of semantic routing and specialization independently of stochastic effects.
- **Stochastic Ensemble (No Iterative Refinement).** Multiple parallel reasoning instances are generated ($N = 5, T = 0.7$) and aggregated via majority voting, while disabling the iterative refinement loop. Comparing this configuration to greedy inference quantifies the contribution of diversity and self-consistency.
- **Full System.** The complete architecture includes semantic routing, stochastic ensemble reasoning, majority voting, and iterative refinement. This configuration represents the final system submitted to the shared task.

Unless otherwise stated, all configurations share the same base prompts and inference constraints, ensuring that observed performance differences are attributable solely to architectural components rather than prompt content.

Table 1

Accuracy (%) on the PFB English Test Set. Comparison of models and inference configurations under the proposed pipeline. Scores are reported as overall accuracy (*Total*) and stratified by the official difficulty tiers (*Easy*, *Medium*, *Hard*); when available, results are also broken down by question typology (*Conceptual*, *Boolean*, *Tabular*). An asterisk (*) marks the configuration submitted to the shared task. The *Top-Competitor System* row reports the official leaderboard reference and is provided only with aggregate and difficulty-level scores; typology-specific results are not available (indicated by “-”).

Model	Setting	Total	Difficulty Level			Question Type		
			Easy	Medium	Hard	Conceptual	Boolean	Tabular
LLaMA 3.1 8B	Baseline	45.15	46.38	45.27	40.62	65.02	31.27	42.51
GPT-OSS-20B	Baseline	66.13	77.11	64.33	59.38	54.32	76.64	69.72
	Single-Pass	77.62	84.94	76.52	71.88	73.93	71.16	88.38
	No Voting	78.52	82.53	77.30	82.81	81.06	68.82	87.20
	No Refinement	88.41	88.55	88.98	81.25	85.76	87.37	92.05
	Full System	87.80	88.10	87.73	87.88	86.93	85.33	90.81
DeepSeek v3.1 671B*	No Refinement	88.41	87.35	88.46	90.62	84.49	88.65	91.77
	Full System*	89.41	87.95	89.62	90.62	86.09	90.00	91.74
Top-Competitor System		89.00	84.00	90.00	87.50	-	-	-

4.2. Performance Analysis

Performance Analysis: English Language. Table 1 reports the results on the English test set, which represents a high-resource and linguistically stable setting. This scenario provides a controlled environment to assess the effectiveness of the proposed inference-time architecture and the contribution of stochastic self-consistency mechanisms, before moving to cross-lingual evaluation.

We begin by analyzing the progressive impact of architectural components on **GPT-OSS-20B**, which serves as the main ablation backbone. The *Baseline* configuration achieves 66.13% accuracy, already substantially outperforming the lightweight LLaMA 3.1 8B baseline (45.15%), but still exhibiting clear weaknesses on structurally complex questions, particularly conceptual and tabular ones. Introducing explicit reasoning via the *Single-Pass* integrated CoT yields a marked improvement (+11.49%), confirming that financial MCQA strongly benefits from structured intermediate reasoning. However, the most

significant gain arises when activating parallel stochastic sampling with majority voting. The *No Refinement* configuration reaches 88.41% accuracy (+10.79% over greedy inference), demonstrating that aggregating multiple reasoning paths effectively suppresses unstable or hallucinatory generations. This effect is especially pronounced for tabular reasoning, where accuracy increases from 88.38% to 92.05%.

The introduction of the iterative refinement mechanism (*Full System*) produces a more nuanced effect. While overall accuracy slightly decreases for GPT-OSS-20B (87.80%), refinement substantially improves performance on hard instances (from 81.25% to 87.88%). This suggests that refinement acts as a targeted intervention for ambiguous or high-complexity cases, at the cost of occasional over-correction on simpler questions. As such, refinement is not uniformly beneficial but proves critical when cognitive load increases.

A breakdown by question typology highlights the role of specialization. Tabular questions consistently achieve the highest accuracy across models, validating the effectiveness of the Analyst prompting strategy based on explicit value extraction and calculation. Boolean verification remains the most challenging category, reflecting the intrinsic difficulty of tracking multiple truth-value dependencies within a single question. Notably, for the full configurations, performance remains stable across difficulty levels, and in some cases improves on the Hard tier, indicating that the architecture does not degrade under increased reasoning complexity.

Regarding model scale, DeepSeek v3.1 is evaluated only under the two most relevant configurations (*No Refinement* and *Full System*). Due to its high computational cost, extensive ablation is impractical; instead, we focus on assessing whether the proposed inference-time orchestration can further enhance an already strong proprietary model. The *Full System* achieves the best overall accuracy (89.41%) and represents our official submission. In contrast, LLaMA 3.1 8B is evaluated exclusively in the baseline setting, as its limited capacity does not support reliable stochastic ensembling, and additional architectural complexity yields marginal returns relative to computational overhead.

Finally, comparison with the official competitive system shows that our approach is particularly effective on edge cases. This indicates that inference-time architectural control can rival, and in some cases surpass, highly specialized baselines, offering a competitive alternative to resource-intensive domain adaptation strategies.

Table 2

Accuracy (%) on the PFB Italian Test Set. Overall accuracy and stratified results by difficulty and question type. The asterisk (*) indicates the official submission. Question-type scores are not available for the Top-Competitor System.

Model	Setting	Total	Difficulty Level			Question Type		
			Easy	Medium	Hard	Conceptual	Boolean	Tabular
LLaMA 3.1 8B	Baseline	40.66	37.95	41.37	39.06	65.79	27.57	32.11
GPT-OSS-20B*	Baseline	54.35	68.67	52.14	43.75	64.47	41.89	59.02
	Single-Pass	72.93	74.10	73.80	59.38	75.66	64.86	79.51
	No Voting	75.62	80.12	75.10	70.31	78.29	67.92	81.90
	No Refinement	86.01	86.75	85.99	84.38	82.57	85.95	89.30
	Full System*	87.71	88.55	87.81	84.38	84.16	88.95	89.60
DeepSeek v3.1 671B	No Refinement	83.80	84.52	84.07	78.79	83.05	83.11	85.14
	Full System	83.92	84.34	83.92	82.81	85.81	83.51	82.62
Top-Competitor System		91.00	86.00	92.00	92.00	-	-	-

Performance Analysis: Italian Language. Table 2 reports the results on the Italian test set. As a mid-resource language in the financial domain, Italian provides a critical benchmark for evaluating cross-lingual transfer and for assessing whether the proposed inference-time architecture can compensate for linguistic and domain mismatches without model retraining. The same models were investigated for English are evaluated here; however, for lightweight models such as LLaMA 3.1 8B, we restrict the

analysis to the monolithic baseline, as preliminary experiments showed that more complex orchestration does not yield stable gains under severe capacity and language constraints.

A direct comparison between English and Italian highlights a pronounced language gap in raw model performance. For GPT-OSS-20B, the monolithic baseline drops from 66.13% in English to 54.35% in Italian, confirming the weaker alignment of general-purpose models with Italian financial language. Crucially, this gap is almost entirely eliminated once the proposed architecture is activated. With the Full System, GPT-OSS-20B reaches 87.71% accuracy, closely matching its English counterpart (87.80%). This result demonstrates that the structured reasoning enforced by semantic routing, role-specific prompting, and self-consistent aggregation transfers effectively across languages, allowing logical scaffolding to compensate for reduced parametric familiarity with Italian.

The Italian setting also provides insight into the interaction between model scale and architectural optimization. While the small LLaMA 3.1 8B model remains below a usable threshold (40.66%), confirming that limited capacity struggles with the combined burden of language and domain complexity, a counterintuitive pattern emerges when comparing larger models. In the Full System configuration, GPT-OSS-20B (87.71%) outperforms the substantially larger DeepSeek v3.1 (83.92%). This suggests that, in non-English settings, effective instruction-following and alignment with structured prompting may outweigh raw parameter count. Consistent with the English results, stochastic ensembling plays a decisive role: transitioning from single-pass greedy inference (No Voting, 75.62%) to parallel stochastic sampling (No Refinement, 86.01%) yields a substantial performance jump, confirming the robustness of the voting mechanism across languages.

A breakdown by question typology reveals a language-specific effect of the Iterative Refinement mechanism. Unlike English, where refinement shows mixed trade-offs, Italian delivers a clear net gain. In particular, Boolean questions benefit markedly, with accuracy increasing from 85.95% to 88.95%. This improvement can be attributed to the syntactic and semantic complexity of Italian financial assertions, where nested clauses and negations increase the risk of brittle reasoning. By masking low-support options and forcing re-evaluation in a constrained space, refinement mitigates linguistic ambiguity and stabilizes logical consistency.

For the Italian track, the submitted system corresponds to the GPT-OSS-20B Full configuration. Compared to the Competitive System, our approach achieves higher accuracy on Easy instances (88% vs. 86%), indicating strong reliability on foundational financial knowledge. However, a gap remains on Medium and Hard questions, where competitors achieve higher scores. This contrast underscores the central trade-off of our approach: while language-specific training can better capture nuanced, culture-dependent financial knowledge, the proposed inference-only architecture attains competitive performance (nearly 88% overall accuracy) without modifying model parameters, highlighting its portability, efficiency, and practical appeal in multilingual settings.

Performance Analysis: Turkish Language Table 3 reports results on the Turkish test set. As a low-resource language for financial NLP, characterized by rich agglutinative morphology and limited domain-specific pre-training data, Turkish represents the most challenging scenario for evaluating cross-lingual generalization and inference stability. The baseline results highlight the severity of this setting. LLaMA 3.1 8B performs close to random guessing (30.87%), confirming that small general-purpose models are unable to cope with the combined linguistic and domain complexity. Even GPT-OSS-20B exhibits a substantial degradation in its monolithic baseline (48.85%), compared to English and Italian. However, once the proposed inference-time architecture is activated, performance improves dramatically. The Full System configuration reaches 88.21% accuracy, yielding a gain of nearly +40 percentage points over the baseline. This result provides the strongest evidence that structured reasoning and controlled orchestration can effectively compensate for limited language-specific pre-training, enabling robust cross-lingual transfer even in highly challenging conditions. The contribution of stochastic aggregation is particularly pronounced in Turkish. Moving from deterministic greedy inference (No Voting, 78.22%) to the stochastic ensemble (No Refinement, 87.01%) yields one of the largest improvements observed across all languages. This confirms that, in high-entropy settings where

individual generations are more prone to local inconsistencies or hallucinations, majority voting acts as an effective stabilizer by filtering idiosyncratic errors across parallel reasoning paths. The effect is especially visible on *Hard* questions, where accuracy drops to 78.12% without refinement but is restored to 87.50% in the Full System. This suggests that, for difficult instances in low-resource languages, the additional constrained re-evaluation step is essential to suppress spurious alternatives and force convergence.

Table 3

Accuracy (%) on the PFB Turkish Test Set. Overall accuracy and results stratified by difficulty level and question typology. The asterisk (*) denotes the official submission. Scores by question type are not available for the Top-Competitor System.

Model	Setting	Total	Difficulty Level			Question Type		
			Easy	Medium	Hard	Conceptual	Boolean	Tabular
LLaMA 3.1 8B	Baseline	30.87	30.72	30.87	31.25	49.50	19.41	26.61
GPT-OSS-20B*	Baseline	48.85	71.08	45.27	34.38	62.54	30.03	58.72
	Single-Pass	72.23	78.92	71.98	57.81	72.94	65.77	78.90
	No Voting	78.22	80.72	77.43	81.25	78.89	72.75	83.74
	No Refinement	87.01	87.95	87.55	78.12	83.44	87.50	89.85
	Full System*	88.21	89.16	88.07	87.50	83.33	89.97	90.80
DeepSeek v3.1 671B	No Refinement	82.42	85.54	82.23	76.56	83.73	77.31	87.16
	Full System	80.60	79.76	80.94	78.79	80.57	76.00	84.57
Top-Competitor System		88.00	88.00	88.00	87.50	-	-	-

A comparison across model scales further reinforces the central role of architectural design. Despite its substantially smaller size, GPT-OSS-20B with the proposed workflow outperforms the much larger DeepSeek v3.1 model (88.21% vs. 80.60%). Moreover, DeepSeek’s performance degrades when the full constraint-based prompting is applied, indicating a potential mismatch between its internal safety or reasoning mechanisms and the strict inference control imposed by our pipeline. These results suggest that, in specialized multilingual financial QA, an inference architecture aligned with the model’s reasoning style can be more impactful than sheer parameter count.

Finally, in the competitive setting, the submitted configuration (GPT-OSS-20B Full System) achieves state-of-the-art performance. It matches or slightly exceeds the average scores of top competitors across all difficulty tiers, reaching 89% on Easy, 88% on Medium, and 87.5% on Hard questions. Notably, this level of performance is obtained without any language-specific fine-tuning or additional data, relying exclusively on inference-time control. In a low-resource scenario such as Turkish, these results demonstrate that structured orchestration and self-consistent aggregation provide a viable and data-efficient alternative to fine-tuning for achieving competitive financial reasoning performance.

Discussion Across Languages and Research Questions. Taken together, the experimental results across English, Italian, and Turkish provide a coherent answer to the four research questions introduced in Section 1. First, results consistently show that inference-time architectural design is sufficient to reach domain, competent performance in financial MCQA (**RQ1**), especially as no model fine-tuning is applied. Across all languages, routing and role-specialized prompting outperform monolithic inference, confirming that explicitly aligning reasoning strategies with question structure is more effective than relying on implicit in-context classification. Second, stochastic self-consistency proves to be a central stabilizing mechanism (**RQ4**). Majority voting over parallel reasoning traces yields substantial gains in all settings, with particularly strong effects for structurally complex questions and in lower-resource languages. Sampling diversity, therefore, is not causing noise, but works as a practical mechanism to foster answer robustness. The iterative refinement step further strengthens this effect by intervening only when consensus is weak, and is especially beneficial for hard instances and linguistically ambiguous cases. Third, cross-lingual transfer is strongly supported by the evidence (**RQ3**). Although raw model

performance degrades when moving from English to Italian and Turkish, the full system largely closes this gap. Once enforced through controlled orchestration, structured reasoning patterns transfer effectively across languages, allowing mid-sized models to approach, or match, performance observed in large-scale resource settings. Finally, analysis across question types and difficulty levels highlights the role of structural complexity (**RQ2**). The system remains robust as difficulty increases, particularly for tabular and hard questions, and benefits from refinement precisely when reasoning ambiguity is highest. Overall, these results indicate that architectural specialization and controlled aggregation matter more than surface form in determining performance stability.

In summary, the experimental evidence demonstrates that constrained agentic prompting and deterministic self-consistency provide a robust, portable, and training-free alternative to model scaling and domain-specific fine-tuning for multilingual financial MCQA.

For a detailed error analysis with representative examples, including cases highlighting the effects of different inference strategies and multilingual reasoning, see Appendix A.2.

Computational Efficiency and Latency Analysis A key advantage of the proposed architecture is its reliance on inference-time optimization combined with efficient model selection. To quantify this, we measured the end-to-end wall-clock time required to process the entire test set (1,001 instances) leveraging the vLLM library⁷.

Benchmarks were conducted on a single NVIDIA A100 GPU (80GB). To simulate a realistic **multi-tenant production environment**, the GPU resources were shared with concurrent active processes. Consequently, the inference engine was strictly capped at a **50GB VRAM budget**. The backbone model, **GPT-OSS-20B**, utilizes a sparse *Mixture-of-Experts* (MoE) architecture. Crucially, despite its 20B total parameter count, the model activates only **~4B parameters per token** during inference.

This architectural sparsity, synergistic with vLLM’s PagedAttention [31], enabled the full agentic workflow (comprising Routing, 5x Stochastic Sampling, and Aggregation) to achieve a total execution time of **929 seconds**, corresponding to an average latency of **~0.92 seconds per query**, fully accounting for the additional refinement rounds.

This result highlights the viability of the approach: by leveraging MoE sparsity, the system delivers the reasoning depth of a large-scale model with the throughput characteristic of a lightweight 4B model, maintaining sub-second latency even under shared resource constraints.

5. Conclusions

This work explored whether general-purpose LLMs can be reliably applied to multilingual Financial MCQA through inference-time control alone. Across English, Italian, and Turkish, we showed that a carefully designed *constrained agentic workflow* is sufficient to achieve strong and stable performance without any form of fine-tuning or parameter updates. The experimental results demonstrate that architectural control at inference time is a powerful alternative to model scaling and language-specific training. By combining semantic routing, role-specialized prompting, and self-consistent aggregation, the proposed system systematically outperforms monolithic prompting strategies and remains robust under increasing structural and linguistic input complexity. In particular, the architecture proves especially effective on tabular and boolean questions, where naive single-pass inference is most prone to instability. A key outcome of this study is the strong cross-lingual behavior of the proposed approach. While raw model performance degrades significantly when moving away from English, the full system largely eliminates this gap. In Italian and Turkish, the constrained agentic pipeline enables a mid-sized open model (GPT-OSS-20B) to reach or surpass the performance of substantially larger proprietary systems. Notably, on Turkish, a low-resource and morphologically complex language, the proposed system matches the best reported results across all difficulty levels, effectively positioning it as one of the strongest systems in the shared task. These findings suggest that, for multilingual financial

⁷<https://github.com/vllm-project/vllm>

reasoning, structured orchestration and controlled self-consistency can act as a practical and cost-efficient substitute for domain-specific fine-tuning. The resulting system is portable, reproducible, and entirely training-free, making it particularly attractive in settings where labeled data or computational resources are limited.

As future work, we plan to explore alternative aggregation strategies. In particular, we aim to replace the current rule-based voting and refinement mechanism with a dedicated adjudication agent (*Judge*), exposed either to the final answers alone or to their associated reasoning traces. This direction may further improve robustness on edge cases while preserving the inference-only nature of the framework.

Acknowledgments

The authors acknowledge financial support from the PNRR MUR project PE0000013-FAIR. Moreover, the authors acknowledge support from Project ECS 0000024 Rome Technopole, - CUP B83C22002820006, NRP Mission 4 Component 2 Investment 1.5, Funded by the European Union - NextGenerationEU.

Declaration on Generative AI

Parts of the writing and editing process for this manuscript were supported by the use of generative AI tools, specifically OpenAI’s ChatGPT and Google’s Gemini. These tools were employed to enhance clarity and correct spelling. All AI-assisted content was carefully reviewed and validated by the authors to ensure accuracy, originality, and compliance with ethical and scientific standards. The authors bear full responsibility for the final content.

References

- [1] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., Chain-of-thought prompting elicits reasoning in large language models, *Advances in neural information processing systems* 35 (2022) 24824–24837.
- [2] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, Y. Zhang, Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. URL: <https://arxiv.org/abs/2303.12712>. *arXiv:2303.12712*.
- [3] Y. Li, S. Wang, H. Ding, H. Chen, Large language models in finance: A survey, in: *Proceedings of the fourth ACM international conference on AI in finance*, 2023, pp. 374–382.
- [4] H. Yang, X.-Y. Liu, C. D. Wang, Fingpt: Open-source financial large language models, 2025. URL: <https://arxiv.org/abs/2306.06031>. *arXiv:2306.06031*.
- [5] F. Borazio, D. Croce, R. Basili, Adapting llms for domain-specific retrieval: A case study in nuclear safety, in: C. Hauff, C. Macdonald, D. Jannach, G. Kazai, F. M. Nardini, F. Pinelli, F. Silvestri, N. Tonellotto (Eds.), *Advances in Information Retrieval*, Springer Nature Switzerland, Cham, 2025, pp. 116–122.
- [6] S. Wu, O. Irsoy, S. Lu, V. Dabrovolski, M. Dredze, S. Gehrmann, P. Kambadur, D. Rosenberg, G. Mann, Bloomberggpt: A large language model for finance, 2023. URL: <https://arxiv.org/abs/2303.17564>. *arXiv:2303.17564*.
- [7] G. Attanasio, P. Basile, F. Borazio, D. Croce, M. Francis, J. Gili, E. Musacchio, M. Nissim, V. Patti, M. Rinaldi, D. Scalena, CALAMITA: Challenge the abilities of LAngeuage models in ITALian, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), *Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024)*, CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 1054–1063. URL: <https://aclanthology.org/2024.clicit-1.116/>.
- [8] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.

- [9] K. Tian, E. Mitchell, H. Yao, C. Manning, C. Finn, Fine-tuning language models for factuality, in: *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*, 2023.
- [10] J. Robinson, C. M. Rytting, D. Wingate, Leveraging large language models for multiple choice question answering, 2023. URL: <https://arxiv.org/abs/2210.12353>. arXiv:2210.12353.
- [11] F. Borazio, A. Shcherbakov, D. Croce, R. Basili, UniTor at BioASQ 2025: Modular Biomedical QA with Synthetic Snippets and Multiple Task Answer Generation, in: G. Faggioli, N. Ferro, P. Rosso, D. Spina (Eds.), *CLEF 2025 Working Notes*, 2025.
- [12] Q. Zhao, M. Zhang, Elimination-based reasoning with llm for multiple-choice educational question answering, *Journal of King Saud University Computer and Information Sciences* 37 (2025) 1–17.
- [13] M. M. Lucas, J. Yang, J. K. Pomeroy, C. C. Yang, Reasoning with large language models for medical question answering, *Journal of the American Medical Informatics Association* 31 (2024) 1964–1975.
- [14] K. Shang, C.-H. Chang, C. C. Yang, Collaboration among multiple large language models for medical question answering, 2025. URL: <https://arxiv.org/abs/2505.16648>. arXiv:2505.16648.
- [15] M. Xue, D. Liu, W. Lei, X. Ren, B. Yang, J. Xie, Y. Zhang, D. Peng, J. Lv, Dynamic voting for efficient reasoning in large language models, in: H. Bouamor, J. Pino, K. Bali (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, Association for Computational Linguistics, Singapore, 2023, pp. 3085–3104. URL: <https://aclanthology.org/2023.findings-emnlp.203/>. doi:10.18653/v1/2023.findings-emnlp.203.
- [16] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, D. Zhou, Self-consistency improves chain of thought reasoning in language models, 2023. URL: <https://arxiv.org/abs/2203.11171>. arXiv:2203.11171.
- [17] Z. Wu, L. Sheng, Y. Xia, Y. Zhang, Y. Chen, A. Zhang, Personalized recommendation agents with self-consistency, in: *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 2978–2982.
- [18] Y. Li, Z. Lin, S. Zhang, Q. Fu, B. Chen, J.-G. Lou, W. Chen, Making large language models better reasoners with step-aware verifier, 2023. URL: <https://arxiv.org/abs/2206.02336>. arXiv:2206.02336.
- [19] H. Park, J. Lee, H. Oh, Fintab-llava: Finance domain-specific table understanding multimodal llm using fintmd, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Springer, 2025, pp. 235–246.
- [20] M. Rizinski, D. Trajanov, Ai agents in finance and fintech: A scientific review of agent-based systems, applications, and future horizons, *Computers, Materials, & Continua* 86 (2026) 1.
- [21] H. Yang, B. Zhang, N. Wang, C. Guo, X. Zhang, L. Lin, J. Wang, T. Zhou, M. Guan, R. Zhang, C. D. Wang, Finrobot: An open-source ai agent platform for financial applications using large language models, 2024. URL: <https://arxiv.org/abs/2405.14767>. arXiv:2405.14767.
- [22] J. Li, J. Zhang, H. Li, Y. Shen, An agent framework for real-time financial information searching with large language models, 2024. URL: <https://arxiv.org/abs/2502.15684>. arXiv:2502.15684.
- [23] S. Ganesh, L. Ardon, D. Borrajo, D. Garg, U. M. Sehwal, A. L. Narayanan, G. Canonaco, M. M. Veloso, Generative ai agents for knowledge work augmentation in finance, *Annual Review of Control, Robotics, and Autonomous Systems* 8 (2025) 189–210.
- [24] H. Du, S. Thudumu, R. Vasa, K. Mouzakis, A survey on context-aware multi-agent systems: Techniques, challenges and future directions, 2025. URL: <https://arxiv.org/abs/2402.01968>. arXiv:2402.01968.
- [25] S. Pateria, B. Subagdja, A.-h. Tan, C. Quek, Hierarchical reinforcement learning: A comprehensive survey, *ACM Computing Surveys (CSUR)* 54 (2021) 1–35.
- [26] I. Okpala, A. Golgoon, A. R. Kannan, Agentic ai systems applied to tasks in financial services: Modeling and model risk management crews, 2025. URL: <https://arxiv.org/abs/2502.05439>. arXiv:2502.05439.
- [27] C. Choi, J. Kwon, A. Lopez-Lira, C. Kim, M. Kim, J. Hwang, J. Ha, H. Choi, S. Yun, Y. Kim, et al., Finagentbench: A benchmark dataset for agentic retrieval in financial question answering, in: *Proceedings of the 6th ACM International Conference on AI in Finance*, 2025, pp. 632–637.
- [28] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, et al., The llama 3 herd of models, 2024. URL: <https://arxiv.org/abs/2407.2>

1783. [arXiv:2407.21783](#).

- [29] S. Agarwal, L. Ahmad, J. Ai, S. Altman, A. Applebaum, E. Arbus, R. K. Arora, Y. Bai, B. Baker, H. Bao, et al., gpt-oss-120b & gpt-oss-20b model card, 2025. URL: <https://arxiv.org/abs/2508.10925>. [arXiv:2508.10925](#).
- [30] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al., Deepseek-v3 technical report, 2025. URL: <https://arxiv.org/abs/2412.19437>. [arXiv:2412.19437](#).
- [31] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, I. Stoica, Efficient memory management for large language model serving with pagedattention, in: Proceedings of the 29th symposium on operating systems principles, 2023, pp. 611–626.

A. Appendix

A.1. Prompt Design and Examples

To operationalize the specialized reasoning capabilities described in Section 3, the architecture relies on a set of strictly formatted prompts. Each prompt is engineered to induce a specific role (Orchestrator, Analyst, Fact-Checker, Researcher) and enforces rigid output constraints (e.g., single-letter parsing). These structural guarantees are essential to enable deterministic question processing and robust aggregation during the voting phase. Representative examples of these prompts are provided below.

Prompt 1: Routing (Orchestrator Module). This prompt activates the Orchestrator, whose role is to classify incoming queries into one of three specialized LLM modules. It ensures that each question is routed to the most suitable reasoning module, **Analyst**, **Fact-Checker**, or **Researcher**, based on its content and structure. By enforcing strict output constraints, the Orchestrator guarantees consistent downstream processing and prevents ambiguity in module selection.

Prompt 1: Routing (Orchestrator Module)

You are the Orchestrator of a financial analysis team.

Your **ONLY** job is to analyze the user input and classify it into one of three categories:

- **ANALYST:** If the input contains a Markdown Table, financial statements, or raw data requiring calculation.
- **FACT_CHECKER:** If the input contains a numbered list of statements (e.g., “1. ... 2. ...”) and asks to identify which are True/False.
- **RESEARCHER:** For standard questions requiring general knowledge, retrieval of financial theory, or conceptual definitions.

Output Constraint: Output **ONLY** the category name: ANALYST, FACT_CHECKER, or RESEARCHER. Do not write anything else.

Prompt 2: Numerical and Table-Based Reasoning (Analyst Module) When financial tables, statements, or explicit numerical constraints are present, this prompt triggers the **Analyst** module. The module extracts required values, performs step-by-step arithmetic verification, and resolves calculation-heavy queries. This structured reasoning approach addresses the combined challenges of table structure, numerical alignment, and domain-specific terminology.

Prompt 2: Numerical and Table-Based Reasoning (Analyst Module)

You are a Quantitative Financial Analyst.

You are provided with financial data (tables, balance sheets, or numerical values) in the prompt.

Strategy:

1. Extract the specific values required from the provided text or table.
2. Perform the necessary arithmetic calculations step-by-step.
3. Compare your result with the options and identify the correct one.

Final Answer: Write **ONLY** the uppercase letter of the chosen option (e.g., A, B, C). Do not include any additional text.

Prompt 3: True/False Verification (Fact-Checker Module) Designed for questions containing multiple independent statements, this prompt engages the **Fact-Checker** module. The system decom-

poses the query, evaluates each statement individually as true or false, and then synthesizes the results to select the correct answer. This strategy ensures high fidelity in Boolean verification tasks.

Prompt 3: True/False Verification (Fact-Checker Module)

You are a Financial Fact-Checker.

The user provided a list of numbered statements. Your goal is to identify which are FALSE (or TRUE, as requested).

Strategy:

1. Decompose: Isolate each numbered statement.
2. Verify each statement independently using your internal knowledge.
3. Explicitly label each statement as TRUE or FALSE.
4. Match: Select the multiple-choice option that corresponds to your findings.

Final Answer: Write ONLY the uppercase letter of the chosen option (e.g., A, B, C). Do not include any additional text.

Prompt 4: Conceptual Domain Knowledge (Researcher Module) This prompt invokes the **Researcher** module to handle conceptual or theoretical finance questions. By leveraging parametric knowledge, the module retrieves relevant definitions, regulatory context, and financial theory to inform qualitative reasoning. This ensures accurate selection of the correct answer based on domain expertise.

Prompt 4: Conceptual Domain Knowledge (Researcher Module)

You are a Financial Academic Researcher.

You are tasked with answering conceptual or theoretical finance questions.

Strategy:

1. Identify the key financial concepts and definitions in the question.
2. Synthesize the relevant domain knowledge required to answer.
3. Select the correct option based on your synthesis.

Final Answer: Write ONLY the uppercase letter of the chosen option (e.g., A, B, C). Do not include any additional text.

A.2. Error Analysis with Representative Examples

To better understand the impact of different prompting, self-consistency, and refinement strategies, we conduct a qualitative error analysis on representative examples where model predictions differ across inference settings. Specifically, we analyze cases in which *Orchestrated Greedy Inference (No Voting)* fails while *Stochastic Ensemble (No Refinement)* succeeds, cases where *our proposed Full System* improves over *Stochastic Ensemble* through *multi-turn refinement*, and a multilingual tabular reasoning case that evaluates the model’s ability to jointly handle non-English (Italian) language understanding, numerical extraction, and table structure. Together, these examples provide insight into the mechanisms driving the observed performance gains.

Case 1 (GPT-OSS-20B): Orchestrated Greedy Inference (No Voting) vs. Stochastic Ensemble (No Refinement) on English Conceptual Question. For this case, we examine a medium-difficulty, conceptual question (ID: Book__219), shown in Table 4. Under the **No Voting** setting, GPT-OSS-20B incorrectly predicts E (“None of the above”). This suggests difficulty in validating all candidate options in a single-pass reasoning process. In contrast, the **Stochastic Ensemble** strategy correctly predicts B with **80% confidence**, supported by the voting distribution [B, B, B, E, B].

Table 4

English conceptual question corrected by Stochastic Ensemble.

Question (ID: Book__219, Medium, conceptual-type, English)

What is a key factor that leads to the linkage between health insurance and employment in the United States?

- A. Government subsidy for health insurance costs
- B. The Stabilization Act of 1942, which exempted insurance from wage controls
- C. Higher premiums for individual health insurance policies
- D. Inability to classify individuals into precise risk groups
- E. None of the above

Correct Answer: B

Interpretation. The “None of the above” option requires global verification of all alternatives rather than local semantic plausibility. While the **No Voting** fails to sufficiently reject plausible distractors, **Stochastic Ensemble** enables historically grounded reasoning to dominate through majority voting.

Case 2 (DeepSeek v3.1 671B): Full System vs. Stochastic Ensemble (No Refinement) on English Boolean Question. For this case, we examine a medium-difficulty, Boolean question (ID: Book__1386), shown in Table 5. Using the **Stochastic Ensemble (No Refinement)**, DeepSeek v3.1 671B predicts **B** with **60% confidence**, based on the vote distribution [D, B, D, B, B]. This outcome reflects competing reasoning paths across self-consistency samples, indicating the absence of a stable consensus in the interpretation of the multi-statement logic.

Table 5

English boolean question corrected by the Full System.

Question (ID: Book__1386, Medium, Boolean-type, English)

1. Hedging is a strategy to increase the returns from foreign investments regardless of currency fluctuations.
2. Foreign tourists benefit from a weaker currency when visiting the country with the stronger currency.
3. When a currency appreciates, it can buy more of other currencies than before.

Which of these statements are **FALSE**?

- A. 2
- B. 1
- C. 2, 3
- D. 1, 2
- E. None of the above

Correct Answer: D

After applying **Full System** with multi-turn refinement, the model correctly predicts **D**, supported by the refined vote distribution [D, D, B, B, D]. In the refinement stage, low-support alternatives are masked, allowing the model to re-evaluate the remaining candidates under a constrained answer space.

Interpretation. The performance improvement from **Stochastic Ensemble (No Refinement)** to **Full System** arises from structured option pruning during refinement. By narrowing the set of plausible answers in the second pass, the model achieves stronger internal agreement and more stable convergence in multi-statement reasoning tasks.

Case 3 (GPT-OSS-20B): Baseline vs. Full System on Italian Tabular Question. To further evaluate the robustness of our approach across *non-English languages* and *tabular reasoning*, we analyze a medium-difficulty, tabular-format question in Italian (ID: FINANCIALS__4501), shown in Table 6.

This example is drawn from the FINANCIALS category and requires precise extraction of numerical values from a multi-column financial statement.

Table 6

Italian tabular question corrected by the Full System according to data derived from Table 7.

Question (ID: FINANCIALS__4501, Medium, Tabular-type, Italian)

Quanto denaro contante era disponibile alla fine del 2023 (USD)?

(Estratto dal Rendiconto Finanziario Consolidato; see Table 7)

A. 50.679.489

B. 36.447.024

C. 39.399.770

D. 69.080.893

E. Nessuna delle precedenti

Correct Answer: A

Under the **GPT-OSS Baseline** setting, the model predicted option **C** that does not correspond to the correct cash position at the end of 2023 in USD, reflecting difficulty in jointly handling table structure, numerical alignment, and Italian-language financial terminology. In contrast, the **Full System** correctly predicts option **A** with **60% confidence**, supported by the voting distribution [C, A, C, A, A].

Interpretation. This example highlights three compounding sources of difficulty: (i) reasoning over a dense financial table, (ii) correct temporal alignment (end of 2023 vs. beginning of the year), and (iii) comprehension of Italian financial terminology such as “Disponibilità liquide e mezzi equivalenti”. The **Baseline** model is distracted by numerically salient but contextually incorrect values (e.g., beginning-of-year figures or KRW-denominated columns).

The **Full System** benefits from multi-sample self-consistency and refinement, enabling consistent alignment between the question constraint (“fine del 2023”, USD) and the correct table cell. Majority voting suppresses spurious but locally plausible alternatives (e.g., option C), while refinement reinforces agreement among samples that correctly track both language and table structure.

Table 7

Estratto del rendiconto finanziario consolidato utilizzato nella domanda FINANCIALS__4501.

Voce	Nota	2024 (KRW)	2023 (KRW)	2024 (USD)	2023 (USD)
Attività di finanziamento					
Aumento netto dei prestiti a breve termine	27	5.871.346	2.145.400	4.307.368	1.573.920
Aumento dei prestiti a lungo termine	27	404.954	354.712	297.084	260.226
Rimborso di obbligazioni e prestiti a lungo termine	27	(1.364.508)	(1.219.579)	(1.001.038)	(894.714)
Dividendi pagati		(10.888.749)	(9.864.474)	(7.988.261)	(7.236.827)
Azioni proprie acquistate		(1.811.775)	–	(1.329.164)	–
Operazioni con azionisti di minoranza		(8.511)	(9.118)	(6.244)	(6.690)
Flussi di cassa netti da attività di finanziamento		(7.797.243)	(8.593.059)	(5.720.255)	(6.304.085)
Riclassificazione in attività destinate alla vendita	32	–	(14.153)	–	(10.383)
Effetto delle variazioni dei tassi di cambio		4.821.010	792.785	3.536.815	581.607
Aumento (diminuzione) netto delle disponibilità liquide		(15.375.314)	19.400.183	(11.279.719)	14.232.465
Disponibilità liquide e mezzi equivalenti					
Inizio dell'esercizio		69.080.893	49.680.710	50.679.489	36.447.024
Fine dell'anno		53.705.579	69.080.893	39.399.770	50.679.489

FadeIT at EVALITA 2026: Overview of the Fallacy Detection in Italian Social Media Texts Task

Alan Ramponi^{1,*}, Sara Tonelli¹

¹Fondazione Bruno Kessler, Digital Humanities Unit – Trento, Italy

Abstract

FADEIT is the first shared task on fallacy detection in social media texts in Italian, an understudied language for this task. FADEIT relies on FAINA, a fallacy detection dataset that includes span-level annotations with overlaps for 20 fallacy types in social media texts about migration, climate change, and public health over a 4-year time period. The shared task is articulated into two subtasks at different granularities: i) *post-level fallacy detection*, aiming at predicting the fallacy types expressed in each input post, and ii) *span-level fallacy detection*, aiming at predicting all text segments expressing any given fallacy type in each input post. Participants' systems are evaluated against two equally valid gold standards (i.e., parallel annotations in FAINA) to account for natural disagreement, in line with recent work advocating the importance of considering human label variation in subjective tasks. FADEIT has attracted wide interest at Evalita 2026 with a total of 25 runs submitted by 7 participant teams. In this paper, we present the task setup, including the data used and the evaluation criteria, as well as the results obtained by all participant teams, an analysis of their approaches, and insights for future research on the topic.

Keywords

Natural language processing, fallacy detection, argumentation mining, human label variation

1. Introduction

Fallacies are arguments that seem valid but are not [1, 2]; namely, statements that are logically flawed or in which evidence is replaced by emotional cues. Fallacious argumentation frequently occurs in everyday discourse, either intentionally – for persuading the audience – or unintentionally. With the widespread use of online platforms, fallacious social media posts have the potential to mislead a large audience, in some cases leading to the proliferation of misinformation [3]. Therefore, recognizing fallacies is paramount not only to limit the spread of misleading content, but also to develop individuals' critical thinking skills and promote democratic debate [4].

Motivated by the intrinsic difficulty of the fallacy detection task for automated systems including large language models (LLMs) [5, 6], we present FADEIT, the first shared task on fallacy detection in Italian social media texts. FADEIT has been organized as part of the Evalita 2026 evaluation campaign [7], and comprises two subtasks of increasing complexity: a subtask that deals with the detection of fallacies at the post level (i.e., a multi-label text classification problem) and a more challenging subtask requiring the detection of fallacies at the span level (i.e., a multi-label span classification problem). FADEIT relies on FAINA [6], the first dataset for fallacy detection in Italian embracing multiple plausible answers and natural disagreement, with annotations across an inventory of 20 fallacy types at the fine-grained level of text segments with potential overlaps. It covers public discourse on migration, climate change, and public health issues in social media posts over a large time frame of 4 years. The evaluation of participants' systems is carried out by comparing their submitted predictions with multiple gold standards – included in the FAINA dataset as parallel annotations – in order to account for human label variation [8] – i.e., the genuine disagreement that naturally occurs in subjective NLP tasks.

In the following sections, we present details on the task (Section 2), the data used for the competition (Section 3), the evaluation setup (Section 4), the participant teams and their results (Section 5), followed by an analysis and discussion (Section 6) and our conclusions (Section 7).

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ alramponi@fbk.eu (A. Ramponi); satonelli@fbk.eu (S. Tonelli)

ORCID 0000-0002-4305-2404 (A. Ramponi); 0000-0001-8010-6689 (S. Tonelli)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Task description

The FADEIT shared task focuses on detecting fallacies expressed in Italian social media texts about migration, climate change, and public health issues at different granularities: at the post level (subtask A; Section 2.1) and at the span level (subtask B; Section 2.2). For each post, there can be zero, one, or more fallacies, among an inventory of 20 fallacy types, to be detected at the chosen granularity. We refer to Section 3.2 for the list of fallacy types and to the FAINA dataset [6] for detailed descriptions.

2.1. Subtask A: Post-level fallacy detection

Given the text of a social media post, predict all the fallacy types expressed in it. This is a multi-label classification task (20 classes) and represents the easiest setup – i.e., there is no need to *locate* each occurring fallacy type within the text, just to *detect* which ones (if any) are expressed in it.

2.2. Subtask B: Span-level fallacy detection

Given the text of a social media post, predict all the text segments expressing fallacies and give each of them a fallacy type. This is a challenging, multi-label span classification task (20 classes) and represents the hardest setup – i.e., different fallacies may overlap, partially or in full, with each other. The text of the posts provided to participants is already divided into tokens.

3. Data

In this section, we summarize how the FAINA dataset used for the FADEIT task has been collected (Section 3.1) and annotated (Section 3.2). Moreover, we provide information on data splits (Section 3.3) and format (Section 3.4). Further details are provided in the original paper introducing the dataset [6].

3.1. Data collection

Data collection was conducted using the Twitter APIs in February 2023.¹ We collected social media posts covering discourse on migration, climate change, and public health using a manually curated list of keywords. The time period of the posts represented in the dataset is from 2019-01-01 to 2022-12-31. From this collection, we selected the posts with at least 5 tokens and greatest potential impact to the society – i.e., those with the highest number of retweets and likes, as in Nakov et al. [9]. We mitigated topic and temporal biases by keeping the top 10 posts for each month and topic combination (e.g., 10 posts about “migration” posted in 2021-01). We further mitigate authors’ stylistic bias by excluding multiple posts by the same user and resampling them until we obtained the 10 posts. As a result, the FAINA dataset consists of 1,440 posts balanced across topics (480 per topic) and time (360 per year).

3.2. Data annotation

The collected posts underwent fine-grained annotation at the span-level with overlaps. Due to the difficulty of the task, we devised an annotation protocol that consisted of five rounds of annotation and discussion among two expert annotators ($\mathcal{A}_1$ and $\mathcal{A}_2$). At each round, each annotator individually located and classified text segments expressing fallacies. Then, annotators met to discuss the instances that diverged in the assigned fallacy type, span extent, or both. Instead of forcing a “single ground truth” in data, the goal of the discussion phase was to minimize annotation errors (e.g., due to attention drops) whilst keeping signals of human label variation (e.g., genuine disagreement, such as multiple plausible annotations due to different interpretations of the text). Overall, FAINA consists of 11,064 annotated spans ($5,532 \pm 253$ /annotator) across 58,490 tokens. For allowing a post-level version of the task to be addressed (i.e., subtask A), span-level annotations have also been transposed to the post-level.²

¹At that time, the Twitter (now X) APIs for research purposes were still available for free.

²This was done by assigning to each post the set of unique fallacy span types occurring in it.

Inventory of fallacy types The 20 fallacy categories are: *Ad hominem* (AH), *Appeal to authority* (AA), *Appeal to emotion* (AE), *Causal oversimplification* (CO), *Cherry picking* (CP), *Circular reasoning* (CR), *Doubt* (DO), *Evading the burden of proof* (EP), *False analogy* (FA), *False dilemma* (FD), *Flag waving* (FW), *Hasty generalization* (HG), *Loaded language* (LL), *Name calling or labeling* (NC), *Red herring* (RH), *Slippery slope* (SS), *Slogan* (SL), *Strawman* (ST), *Thought-terminating cliché* (TC), and *Vagueness* (VA). For fallacy definitions, inter-annotator agreement scores before and after discussions, annotation guidelines, and statistics, please refer to the paper introducing FAINA [6]. We provide examples from FAINA in Figure 1.

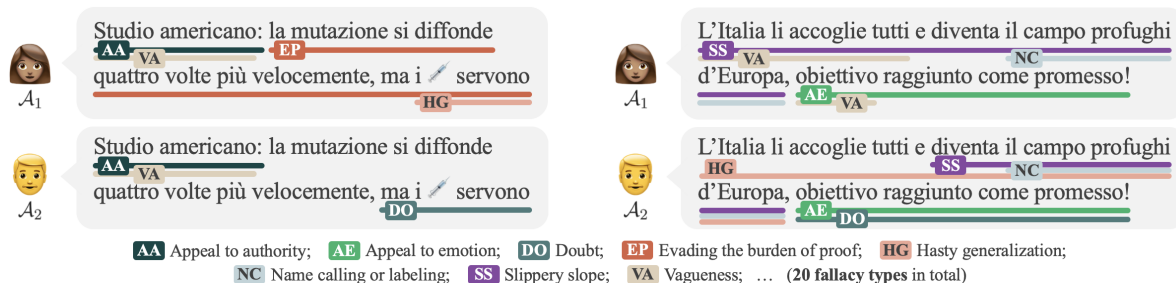


Figure 1: Example of annotated posts from the FAINA dataset [6], keeping genuine disagreement among two expert annotators ($\mathcal{A}_1$, $\mathcal{A}_2$). English translations: “American study: mutation spreads four times faster, but *are needed*” (left); “Italy welcomes them all and becomes Europe’s refugee camp, goal achieved as promised!” (right).

3.3. Data splits

For the purpose of the shared task, the dataset has been split into two official sets: one for training/development (80%; 1,152 posts)³ and one for testing (20%; 288 posts). These have been created by paying particular attention to label, time, and topic distribution across the splits to ensure reliability in the official evaluation. The posts represented in the splits are the same for both subtask A and B.

3.4. Data format

Subtask A For *post-level fallacy detection*, the data is in a tab-separated format with a header line. Each line consists of information about each post (i.e., id, date, topic, text, labels). Post-level annotations by each annotator are provided in separate columns and multiple annotations for the same post and annotator are separated by a pipe. Specifically, each post is represented as shown in Table 1 (*top*).

Subtask B For *span-level fallacy detection*, the data format is based on the CoNLL format. Each post is separated by a blank line and consists of a header with post information, followed by each token in the text (with tab-separated information) separated by newlines. Token annotations follow the BIO scheme (i.e., B: begin, I: inside, O: outside) and multiple annotations for the same token and annotator are separated by a pipe. Specifically, a post in FAINA is represented as shown in Table 1 (*bottom*).

Table 1

FAINA format for subtask A (*top*) and subtask B (*bottom*). Each variable is described in Section 3.4.

\$POST_ID	\$POST_DATE	\$POST_TOPIC	\$POST_TEXT	\$LABELS_BY_ANN_1	\$LABELS_BY_ANN_2
# post_id = \$POST_ID # post_date = \$POST_DATE # post_topic_keywords = \$POST_TOPIC # post_text = \$POST_TEXT \$TOKEN_1 \$TOKEN_1_TEXT \$TOKEN_1_LABELS_BY_ANN_1 \$TOKEN_1_LABELS_BY_ANN_2 ... \$TOKEN_N \$TOKEN_N_TEXT \$TOKEN_N_LABELS_BY_ANN_1 \$TOKEN_N_LABELS_BY_ANN_2					

³Participant teams have been left free to decide how to split the training/development set to tune and select their systems.

The variables in Table 1 are defined as follows:

- **\$POST_ID**: the identifier of the post, different from the Twitter one to preserve user’s anonymity;
- **\$POST_DATE**: the date of the post (in the YYYY-MM format);
- **\$POST_TOPIC**: the topic of the post (i.e., “migration”, “climate change”, or “public health”);
- **\$POST_TEXT**: the text of the post, anonymized with placeholders;⁴
- **\$LABELS_BY_ANN_j**: the fallacy label(s) assigned by annotator j for the post (e.g., “Vagueness”, “Strawman”). In the case where multiple labels for the post are assigned by the same annotator j , these are separated by a pipe and ordered lexicographically, e.g., “Strawman|Vagueness”. In the case where no labels for the post are assigned by the same annotator j , the label is empty;
- **\$TOKEN_i**: the index of the token within the post (i.e., an incremental integer);
- **\$TOKEN_i_TEXT**: the text of the i -th token within the post;
- **\$TOKEN_i_LABELS_BY_ANN_j**: the fallacy label(s) assigned by annotator j for the i -th token within the post. Each label follows the format \$BIO-\$LABEL, where \$BIO is the BIO tag and \$LABEL is the fallacy label (e.g., “Vagueness”, “Strawman”), e.g., “B-Vagueness”, “I-Strawman”, and “O”. In the case where multiple labels for the i -th token are assigned by the same annotator j , these are separated by a pipe and ordered lexicographically by \$LABEL, e.g., “I-Strawman|B-Vagueness”.

We left it up to participant teams to decide whether to aggregate gold annotations by different annotators (e.g., using majority voting), using only one, or leveraging all of them for designing their systems. Nevertheless, to account for human label variation, systems are evaluated against all gold standards (Section 4.1). An example of a post following the aforementioned data formats is presented in Table 2.

Table 2

Post from Figure 1 (*left*) following the data format for subtask A (*top*) and B (*bottom*). In both cases, the last two columns indicate annotations provided by annotators $\mathcal{A}_1$ and $\mathcal{A}_2$ due to different (equally valid) interpretations.

658	2021-06	public health	Studio americano: la mutazione si diffonde quattro volte più velocemente, ma i  servono	Appeal-to-authority Evading-the-burden-of-proof Hasty-generalization Vagueness	Appeal-to-authority Doubt Vagueness
# post_id = 658					
# post_date = 2021-06					
# post_topic_keywords = public health					
# post_text = Studio americano: la mutazione si diffonde quattro volte più velocemente, ma i  servono					
1	Studio			B-Appeal-to-authority B-Vagueness	B-Appeal-to-authority B-Vagueness
2	americano			I-Appeal-to-authority I-Vagueness	I-Appeal-to-authority I-Vagueness
3	:			I-Appeal-to-authority	I-Appeal-to-authority
4	la			B-Evading-the-burden-of-proof	O
5	mutazione			I-Evading-the-burden-of-proof	O
6	si			I-Evading-the-burden-of-proof	O
7	diffonde			I-Evading-the-burden-of-proof	O
8	quattro			I-Evading-the-burden-of-proof	O
9	volte			I-Evading-the-burden-of-proof	O
10	più			I-Evading-the-burden-of-proof	O
11	velocemente			I-Evading-the-burden-of-proof	O
12	,			I-Evading-the-burden-of-proof	O
13	ma			I-Evading-the-burden-of-proof	B-Doubt
14	i			I-Evading-the-burden-of-proof B-Hasty-generalization	I-Doubt
15				I-Evading-the-burden-of-proof I-Hasty-generalization	I-Doubt
16	servono			I-Evading-the-burden-of-proof I-Hasty-generalization	I-Doubt

4. Evaluation

Each team was allowed to submit up to 3 runs (i.e., predictions on the test set) for each subtask. We here introduce the metrics used for assessing performance (Section 4.1) and our baselines (Section 4.2).

4.1. Metrics

We employ different metrics for evaluating participants’ runs in subtask A and B, as detailed below.

⁴User mentions, URLs, email addresses, and phone numbers are replaced with [USER], [URL], [EMAIL], and [PHONE] placeholders, respectively.

Subtask A The submitted runs are evaluated using micro- and macro-averaged precision, recall, and F_1 score, averaged on the two equally-valid gold standard annotations of the 20% held-out test set. Runs are then ranked by micro F_1 score.

Subtask B We evaluate the runs using metrics designed for span-level annotations with potential overlaps, averaged on the two equally-valid gold standard annotations of the 20% held-out test set. We adopt micro- and macro-averaged precision, recall, and F_1 score variants proposed by Da San Martino et al. [10], extended to work at the token level. Partial credit is therefore given to partial span matches, proportional to the length of the match in terms of tokens. To account for the severity of labeling errors (e.g., predicting *Red herring* instead of *Appeal to authority* is less problematic than predicting *False dilemma*), results are also computed in a “soft” evaluation mode, namely giving partial credit (i.e., 0.5 instead of 1.0) if the predicted label is an immediate parent of the actual label in the taxonomy of fallacy types by Ramponi et al. [6]. Runs are then ranked by micro F_1 score in the *soft* evaluation mode.

4.2. Baselines

As baseline systems, we provided two encoder-based models for each subtask. These models have been previously described in the paper introducing FAINA [6] and are summarized in the following.⁵

MVML-ALB and MVML-UMB models A multi-view, multi-label (MVML) model that relies on a shared encoder (either ALBERTo [11] or UmBERTo [12], i.e., ALB or UMB), uses $D = |A|$ decoders (one for each annotation view, i.e., for the labels assigned by each annotator), and outputs D sets of predicted labels containing all fallacy labels that exceed a threshold τ (with $\tau = 0.7$).

MVMD-ALB and MVMD-UMB models A multi-view, multi-decoder (MVMD) model that relies on a shared encoder (either ALBERTo [11] or UmBERTo [12], i.e., ALB or UMB), uses a separate decoder for each annotation view A and fallacy type F (i.e., $D = |A \times F|$), and outputs D sets of predicted labels (i.e., either ‘B’, ‘I’, or ‘O’ for each fallacy label and annotation view). All decoders are given equal importance in the computation of the multi-task learning loss.

5. Participants and results

The FADEIT shared task has attracted a total of 25 runs by 7 participant teams. Specifically, for subtask A we received 16 runs by 6 teams,⁶ whereas for the more challenging subtask B we received 9 runs by 3 teams. Overall, FADEIT has been one of the most participated shared tasks at Evalita 2026 and attracted interest of teams from both academia and industry, representing institutions across five different countries (i.e., India, Italy, Japan, Netherlands, and Vietnam). An overview of participant teams’ approaches is provided in Section 5.1, whereas results for both subtasks are presented in Section 5.2.

5.1. Overview of participant teams’ approaches

A summary of the approaches adopted by each team is provided in the following, alphabetically ordered by team name. For additional details on each submitted run (e.g., model versions, hyper-parameter choices, data and prompt variations), we refer the reader to the system description paper of each team.

Kenji-Endo [13] The team proposed a model with focus on efficiency to tackle multiple Evalita tasks, including FADEIT’s subtask A. The system is based on a decoder-only causal language model that was first pretrained on a mixture of Italian corpora, and then fine-tuned on FAINA data in a discriminative setting by taking into account class imbalance in the loss function. The team submitted a single run.

⁵The code for the baselines is available in the original FAINA’s repository: <https://github.com/dhfbk/faina>.

⁶These include a late run by the team *MALTO* (run 3).

Label [14] The team participated in subtask A and experimented with a two-step prompting approach using LLMs – namely, by first generating a text discussing the potential presence of each fallacy type and then producing a score indicating the likelihood of each of them being present, based on both the generated analysis and the original text of the post. The scores are either used to determine if a fallacy type has to be outputted (based on a tuned threshold value; run 3) or as features, together with topic information and statistical features about the distribution of scores across fallacy types, for training a multi-layer perceptron that models and predicts individual annotators’ labels (run 1 and 2).

MALTO [15] The team participated in subtask A by fine-tuning an encoder-based model pretrained on Italian data using a global multi-label classification threshold. The submitted runs reflect different hyper-parameter configurations and data composition for fine-tuning. Specifically, in run 2, the team used a binary cross-entropy loss without class weights, whereas in run 3, they used a binary cross-entropy loss with positive class weights and paraphrase-based augmented data for fine-tuning.⁷

PuDy [16] The team participated in subtask B. They adopted a span-based modeling approach using an encoder-based model and representing spans of up to 20 tokens as the concatenation of boundary span embeddings and a learned span length embedding. The system employs a hierarchy-based label propagation strategy, making fallacy types that are immediate parents of the gold fallacy sub-types to receive supervision. Moreover, the surrounding context of fallacious spans is perturbed using an LLM (in a 2-shot setting) to reduce overfitting to contextual lexical cues. The three runs reflect different hyper-parameter and data configurations of the same system.

RBG-AI [17] The team participated in both subtasks with three runs each using a unified prompt-based framework. They tested instruction-tuned LLMs in few-shot settings (namely, using 3-, 5-, and 10-shots), selecting examples that maximize diversity in terms of fallacy types, multi-label instances, and annotation views. During prediction, outputs are bounded to coarse-grained groups of fallacy types derived from semantic closeness and corpus-level distributional statistics of FAINA’s fallacy inventory.⁸

TiGRO [18] The team submitted three runs for each of the subtasks. For subtask A, they experimented with a one-vs-rest strategy by fine-tuning 20 binary classifiers – one per fallacy type – using a multilingual encoder (run 1), and with a multi-task learning approach, using an encoder pretrained on Italian data and a decoder per fallacy type, either considering post-level annotations (run 2) or jointly accounting for post- and span-level labels using a total of 40 decoders (run 3). For subtask B, the team employed a multi-task learning approach with 20 span-level decoders and an encoder pretrained on Italian data (run 1), employed the same system used for run 3 in subtask A (run 2), and experimented with a variant of this system by substituting the backbone encoder with a multilingual one (run 3).

UNICA [19] The team participated in subtask A and proposed different approaches based on fine-tuning and retrieval-augmented generation (RAG). All approaches used training data that was previously augmented via LLM prompting and back-translation. In run 1, a closed-weight LLM from the OpenAI family was instructed with few-shot examples. The examples were dynamically selected based on their semantic similarity with the input text and through RAG, using a closed-weight text embedding model from the same family. In run 2 and 3, fine-tuning of LLMs was conducted using models from the Gemma and Mixtral families, respectively.

5.2. Results

In this section, we provide the results on the official test set for all runs submitted by participant teams in the post-level (Section 5.2.1) and span-level (Section 5.2.2) fallacy detection setups.

⁷At the time of writing, we do not have details about run 1; we refer the reader to the MALTO’s technical report.

⁸At the time of writing, we do not have details on the exact approach adopted in each run; we refer to the RBG-AI’s report.

5.2.1. Subtask A: Post-level fallacy detection

The results on the test set for all the runs submitted by teams participating in subtask A are reported in Table 3. By looking at micro-averaged F_1 scores, the *TiGRO* team achieves the best results on the task (56.39 micro F_1 ; run 3) by modeling post- and span-level annotations jointly in a multi-task learning framework. *MALTO* follows it with a system that uses data augmentation via paraphrasing and takes into account class imbalance in the loss computation (54.63 micro F_1 ; run 2). The other runs by *TiGRO* place third (52.41 micro F_1 ; run 2) and fifth (48.65 micro F_1 ; run 1), with a multi-task learning approach considering post-level annotations and by employing a one-vs-rest strategy, respectively. Interestingly, all these systems do not rely on decoder-based LLMs but on encoder-based models, confirming what has been observed in previous work about the challenges of this task for LLMs [5, 6]. The submitted run that ranks the highest among those using LLMs is run 1 by *UNICA* (49.71 micro F_1). It places fourth by instructing models from the OpenAI family in a few-shot manner, using examples that were dynamically selected using semantic similarity and retrieval-augmented generation methods.

Table 3

Test set results for subtask A (*post-level fallacy detection*). P: Precision, R: Recall, F_1 : F_1 score. All scores are reported in their micro- and macro-averaged flavors. Runs are ranked by decreasing micro-averaged F_1 score, and the best score for each metric is in bold. Baselines are highlighted in yellow. ‘*’ indicates late submissions.

	Team	Run	micro-averaged			macro-averaged		
			P	R	F_1	P	R	F_1
1	TiGRO	3	53.24	59.99	56.39	34.47	34.74	33.35
2	MALTO	2	55.75	53.60	54.63	41.95	30.95	32.63
3	TiGRO	2	53.43	51.48	52.41	35.91	27.21	27.94
4	UNICA	1	55.22	45.23	49.71	47.13	37.28	36.75
5	TiGRO	1	62.52	39.85	48.65	27.46	18.11	20.57
6	UNICA	3	51.19	44.26	47.45	35.08	24.95	26.14
7	Kenji-Endo	1	49.38	45.55	47.37	14.25	17.17	13.71
8	UNICA	2	58.70	38.44	46.44	35.68	19.53	22.76
9	MALTO	1	46.56	43.65	45.05	28.11	23.33	23.17
<i>MVML-ALB</i>			<i>64.29</i>	<i>34.41</i>	<i>44.82</i>	<i>37.80</i>	<i>15.42</i>	<i>19.68</i>
10	RBG-AI	2	33.09	57.78	42.07	26.54	46.21	29.32
*	MALTO	3	37.45	45.04	40.88	23.53	30.02	25.64
11	RBG-AI	3	30.65	57.62	40.00	31.08	54.90	31.31
12	Label	3	27.60	68.08	39.26	22.01	56.97	29.04
13	Label	1	52.82	30.94	38.96	14.50	10.13	10.31
14	RBG-AI	1	36.35	41.11	38.57	32.77	29.60	23.42
15	Label	2	52.76	30.11	38.32	14.62	10.17	10.82
<i>MVML-UMB</i>			<i>38.53</i>	<i>14.28</i>	<i>20.84</i>	<i>15.13</i>	<i>3.45</i>	<i>5.10</i>

In contrast, when looking at macro-averaged scores, *UNICA* ranks first (36.75 macro F_1 ; run 1). By looking at the per-fallacy scores (Figure 2), *UNICA* (run 1) obtains more balanced scores across fallacy types compared to other teams that ranked higher according to the micro F_1 metric. The high macro F_1 score obtained by the system could therefore be attributed to the good performance achieved on fallacy types that are under-represented in the data (e.g., *Causal oversimplification*, *Slippery slope*), which fine-tuned encoder-based models typically struggle to capture due to the limited number of instances available for training. Nevertheless, the winning *TiGRO* run and the run by *MALTO* that ranked second still take the second (33.35 macro F_1 ; run 3) and the third (32.63 macro F_1 ; run 2) place when looking at macro F_1 scores, respectively, indicating high robustness and showing good performance for most under-represented classes (see Figure 2). Finally, the competitive performance of *RBG-AI* (31.31 and 29.32 macro F_1 ; run 3 and 2) and *Label* (29.04 macro F_1 ; run 3) in terms of macro-averaged scores, despite placing eleventh, tenth, and twelfth according to micro F_1 , respectively, is likewise attributable to the good performance on minority fallacy categories and little performance degradation on majority labels

such as *Loaded language* and *Name calling or labeling*. All runs outperform the MVML-UMB baseline, whereas MVML-ALB is still competitive, especially when looking at micro-averaged scores.

Further details on per-class scores obtained by participant teams' runs can be found in Figure 2.

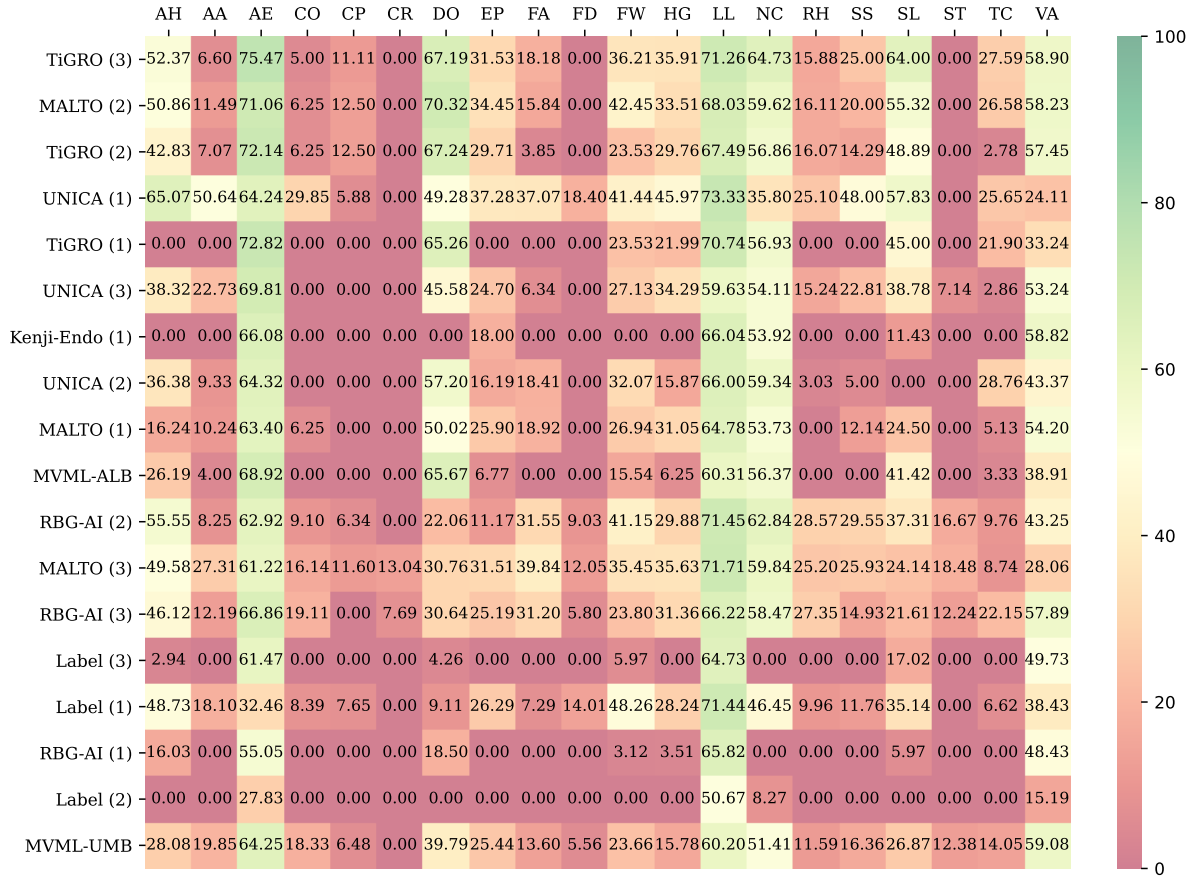


Figure 2: Test set results divided by fallacy type for subtask A (*post-level fallacy detection*) in terms of F_1 score. Participant teams (with run numbers within parentheses) are on the rows, and fallacy types are on the columns. AH: Ad hominem; AA: Appeal to authority; AE: Appeal to emotion; CO: Causal oversimplification; CP: Cherry picking; CR: Circular reasoning; DO: Doubt; EP: Evading the burden of proof; FA: False analogy; FD: False dilemma; FW: Flag waving; HG: Hasty generalization; LL: Loaded language; NC: Name calling or labeling; RH: Red herring; SS: Slippery slope; SL: Slogan; ST: Strawman; TC: Thought-terminating cliché; VA: Vagueness.

5.2.2. Subtask B: Span-level fallacy detection

Test set results for all runs submitted for subtask B are shown in Table 4. All teams outperform the MVMD-UMB baseline, but only the *TiGRO* team achieves higher performance than MVMD-ALB in all runs in the *strict* mode. According to the official shared task metric (micro-averaged span-level F_1 score in the *soft* mode), all the runs by the *PuDy* team ranked first, followed by those by *TiGRO* and those by *RBG-AI*. Specifically, the best system (50.92 micro F_1 , *soft*; *PuDy*, run 1) uses a span-based modeling approach and relies on the UmBERTo encoder-based model. By looking at micro-averaged scores in the *strict* mode (i.e., when requiring the prediction of the exact fallacy types, without granting partial scores for non-severe errors, see Section 4.1), we observe that the best system is the multi-task model with 40 decoders and mmBERT as encoder by *TiGRO* (42.13 micro F_1 , *strict*; run 3). According to this metric, this *TiGRO* system outperforms the scores of the best *PuDy* system (31.97 micro F_1 , *strict*; run 1). When looking at macro-averaged scores, *TiGRO* achieves the best results (26.05 macro F_1 , *strict*; run 1), as also shown by individual fallacy scores in Figure 3. Overall, we observe that systems by both *PuDy* and *TiGRO* are competitive and can be used for different use cases. For instance, if we have no strict

requirements on the identification of exact fallacy categories, *PuDy*'s one would be the system to go. If we are interested in recognizing precise fallacy types, one would prefer the *TiGRO* system instead.

More in general, the choice of a system often depends on the fallacy types of interest. Further details on the scores for each fallacy type obtained by participant teams' runs can be found in Figure 3.

Table 4

Test set results for subtask B (*span-level fallacy detection*). P: Precision, R: Recall, F_1 : F_1 score (span-level variants). All scores are reported in both *strict* and *soft* evaluation modes in their micro- and macro-averaged flavors, where applicable. Runs are ranked by decreasing micro-averaged F_1 score in the *soft* evaluation mode, and the best score for each metric is in bold. Baselines are highlighted in yellow.

Team	Run	STRICT MODE						SOFT MODE			
		micro-averaged			macro-averaged			micro-averaged			
		P	R	F ₁	P	R	F ₁	P	R	F ₁	
1	PuDy	1	27.68	37.83	31.97	26.24	22.39	21.11	43.76	60.88	50.92
2	PuDy	2	25.96	40.80	31.73	19.24	23.73	18.95	40.30	64.79	49.69
3	PuDy	3	29.90	32.96	31.36	20.99	18.83	18.20	45.75	52.36	48.83
4	TiGRO	3	47.82	37.67	42.13	35.69	23.23	25.79	51.01	40.25	44.98
5	TiGRO	1	38.33	40.35	39.30	31.52	25.30	26.05	42.50	45.20	43.80
6	TiGRO	2	38.23	40.05	39.11	29.85	24.16	24.76	42.68	44.87	43.74
MVMD-ALB			48.83	26.87	34.66	36.13	16.42	20.87	52.98	29.48	37.89
7	RBG-AI	3	19.47	25.25	21.99	17.68	16.43	15.18	24.18	32.44	27.71
8	RBG-AI	2	19.21	18.24	18.71	17.52	12.98	12.66	24.67	23.96	24.31
9	RBG-AI	1	17.13	11.01	13.41	16.66	7.67	9.55	20.70	13.27	16.17
MVMD-UMB			60.94	3.05	5.80	10.51	3.21	3.80	65.97	3.28	6.25

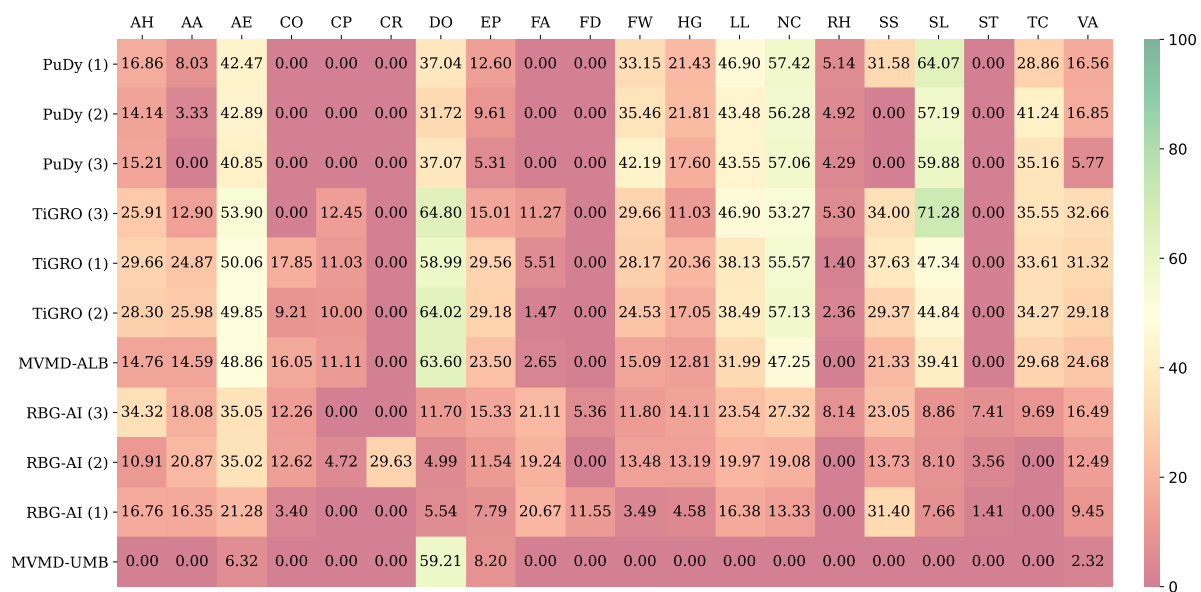


Figure 3: Test set results divided by fallacy type for subtask B (*span-level fallacy detection*) in terms of F_1 score (span-level variant, *strict* evaluation mode). Participant teams (with run numbers within parentheses) are on the rows, and fallacy types are on the columns. AH: Ad hominem; AA: Appeal to authority; AE: Appeal to emotion; CO: Causal oversimplification; CP: Cherry picking; CR: Circular reasoning; DO: Doubt; EP: Evading the burden of proof; FA: False analogy; FD: False dilemma; FW: Flag waving; HG: Hasty generalization; LL: Loaded language; NC: Name calling or labeling; RH: Red herring; SS: Slippery slope; SL: Slogan; ST: Strawman; TC: Thought-terminating cliché; VA: Vagueness.

6. Analysis and discussion

Models All participant teams used transformer-based language models as part of their systems. Among encoder-based models, *MALTO* used ALBERTo [11], *PuDy* employed UmBERTo [12], and *TiGRO* used both ALBERTo and mmBERT [20] in their runs. As regards decoder-based models, open-weight LLMs such as Gemma 3 12B [21] and LLaMa 3.1 8B [22] have been used by *Label* and *RBG-AI* teams, respectively, as well as Mixtral 8x7B [23] and Gemma 3 12B [21] by *UNICA*. Closed-weight models have been used by *UNICA* (i.e., GPT-5, GPT-5.1, and text-embedding-3-small), *PuDy* employed Gemini [24] for the contextual enrichment phase of their systems, whereas *MALTO* used ChatGPT for paraphrase-based data augmentation. *TiGRO* is the only team that used multi-task learning in their runs by leveraging the MaChAmp toolkit [25], showing improvements in performance compared to a single task setup. Finally, the *Kenji-Endo* team trained a causal language model based on the Qwen3 architecture [26]. Given the different strengths of encoder- and decoder-based models in the task (Section 5.2), studying the interplay among them is a valuable direction for future work.

Human label variation and extra-linguistic information Although we provide the training/development set with parallel annotators’ labels as well as topic and time period metadata for each post, this information has not been extensively leveraged by participant teams. The only exception is the *Label* team, that explicitly used the genuine disagreement in the FAINA data for training their classifier and predicting annotator-specific fallacy labels. They also experimented by using topic information, demonstrating that it is a useful feature for fallacy classification. We expect that future work will explore more approaches in this direction, embracing both human label variation [8] and addressing out-of-distribution generalization [27] by leveraging topic and time period information in FAINA data.

Data augmentation Three teams used data augmentation strategies: *MALTO*, *PuDy*, and *UNICA*. Specifically, *MALTO* (run 3) employed ChatGPT to generate paraphrases of selected training data instances (i.e., those with fallacy labels appearing < 100 times in the training set annotations by $\mathcal{A}_1$) by preserving the same post-level labels. Along with original data instances, they then used augmented posts to fine-tune their model for run 3. *PuDy* used Gemini for perturbing the context of the span (i.e., the text before and after it) to be classified. While *PuDy*’s approach seems promising for our task, *MALTO*’s one led to performance degradation. As noted by *MALTO*, their approach is likely to introduce label noise, affecting learning. To increase the chance that labels associated to generated paraphrases are still applicable, in future work a classifier trained on gold data can be used for further validation, as previously done for other tasks such as hate speech detection [28, 29]. Finally, all runs by *UNICA* used augmented training data obtained by instructing GPT-5.1 to generate additional examples for minority fallacy types (i.e., 50 posts for each fallacy type appearing $< 3\%$ of the times in the original training set). Moreover, they further augmented training data by back-translating – with Spanish and French as pivot languages – a random subset of the training data using OPUS-MT translation models [30], discarding augmented instances that exhibited a cosine similarity $< 75\%$ compared to their original counterparts.

Overall, data augmentation leads to mixed results and its efficacy depends on the specifics of each approach. Besides augmentation, future work can consider to enrich existing instances with additional layers of information to be leveraged, such as check-worthiness [31] and argumentation schemes [32].

7. Conclusions

This paper provided an overview of FADEIT, the first shared task on fallacy detection in Italian social media posts organized as part of Evalita 2026. FADEIT attracted notable interest from the research community, registering a total of 25 submitted runs by 7 participant teams from institutions across five different countries. The results of the shared task and our analysis suggest that there is still ample room for improvement in fallacy detection performance, especially for the more challenging yet analytically

useful span-level setup. We hope that our shared task, the dataset, and the evaluation protocol will foster further research in fallacy detection with human label variation.

Acknowledgments

This work has been funded by the European Union’s Horizon Europe research and innovation programmes under grant agreement No. 101070190 (AI4Trust) and under the Marie Skłodowska-Curie grant agreement No. 101073351 (HYBRIDS).

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] C. L. Hamblin, *Fallacies*, Advanced Reasoning Forum, Socorro, USA, 2022.
- [2] C. W. Tindale, *Fallacies and Argument Appraisal*, Critical Reasoning and Argumentation, Cambridge University Press, Cambridge, UK, 2007. doi:<https://doi.org/10.1017/CBO9780511806544>.
- [3] E. Musi, M. Aloumpi, E. Carmi, S. Yates, K. O’Halloran, Developing fake news immunity: Fallacies as misinformation triggers during the pandemic, *Online Journal of Communication and Media Technologies* 12 (2022) e202217. doi:<https://doi.org/10.30935/ojcm/12083>.
- [4] U. Ecker, J. Roozenbeek, S. van der Linden, L. Q. Tay, J. Cook, N. Oreskes, S. Lewandowsky, Misinformation poses a bigger threat to democracy than you might think, *Nature* 630 (2024) 29–32. doi:<https://doi.org/10.1038/d41586-024-01587-3>.
- [5] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 8172–8187. URL: <https://aclanthology.org/2022.emnlp-main.560/>. doi:10.18653/v1/2022.emnlp-main.560.
- [6] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.
- [7] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, EVALITA 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for Italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [8] B. Plank, The “problem” of human label variation: On ground truth in data, modeling and evaluation, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 10671–10682. URL: <https://aclanthology.org/2022.emnlp-main.731/>. doi:10.18653/v1/2022.emnlp-main.731.
- [9] P. Nakov, A. Barrón-Cedeño, G. Da San Martino, F. Alam, R. Míguez, T. Caselli, M. Kutlu, W. Zaghoulani, C. Li, S. Shaar, H. Mubarak, A. Nikolov, Y. S. Kartal, Overview of the CLEF-2022 CheckThat! lab task 1 on identifying relevant claims in tweets, in: *Proceedings of the Working Notes of CLEF 2022 - Conference and Labs of the Evaluation Forum*, CEUR-WS.org, 2022. URL: <https://ceur-ws.org/Vol-3180/paper-28.pdf>.

- [10] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, P. Nakov, Fine-grained analysis of propaganda in news articles, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5636–5646. URL: <https://aclanthology.org/D19-1565/>. doi:10.18653/v1/D19-1565.
- [11] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, V. Basile, ALBERTo: Italian BERT language understanding model for NLP challenging tasks based on tweets, in: R. Bernardi, R. Navigli, G. Semeraro (Eds.), Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019), CEUR Workshop Proceedings, Bari, Italy, 2019, pp. 312–317. URL: <https://aclanthology.org/2019.clcit-1.47/>.
- [12] L. Parisi, S. Francia, P. Magnani, UmBERTo: An Italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020. Accessed: 2026-01-01.
- [13] C. J. Scozzaro, M. Rinaldi, G. Mittone, M. A. Stranisci, Kenji-Endo: a BabyLM @EVALITA, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [14] T. Labruna, E. Papadopoulos, Label at FadeIT: Fallacy-aware LLM reasoning for score-based classification, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [15] M. Salami, L. M. Rodia, V. Schiau, E. A. Munis, C. Savelli, F. Giobergia, MALTO at FadeIT: A BERT-based system for multi-label fallacy detection in Italian social media, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [16] D. D. Phu, S. B. Hong, T. V. Dang, PuDy at FadeIT: Enhancing fine-grained fallacy detection with hierarchy-aware training and contextual enrichment, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [17] Meenakshi, J. R, R. U, B. G. HB, M. Ptaszynski, RBG-AI at FadeIT: Prompted LLMs with label abstraction for logical fallacy detection, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [18] S. Atzeni, G. Sarti, T. Caselli, M. Nissim, TiGRO at FadeIT: E pluribus unum – A multi-task approach to fallacy detection and span identification, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [19] M. Fenu, M. Atzori, Unica at FadeIT: Adapting large language models to fallacy identification in social networks, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [20] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, B. V. Durme, mmBERT: A modern multilingual encoder with annealed language learning, arXiv preprint arXiv:2509.06888 (2025). URL: <https://arxiv.org/abs/2509.06888>.
- [21] Gemma Team, et al., Gemma 3 technical report, arXiv preprint arXiv:2503.19786 (2025). URL: <https://arxiv.org/abs/2503.19786>.
- [22] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, et al., The Llama 3 herd of models, arXiv preprint arXiv:2407.21783 (2024). URL: <https://arxiv.org/abs/2407.21783>.
- [23] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. de las Casas, E. B. Hanna, F. Bressand, G. Lengyel, G. Bour, G. Lample, L. R. Lavaud, L. Saulnier, M.-A. Lachaux, P. Stock, S. Subramanian, S. Yang, S. Antoniak, et al., Mixtral of experts, arXiv preprint arXiv:2401.04088 (2024). URL: <https://arxiv.org/abs/2401.04088>.

- [24] Gemini Team Google, et al., Gemini: A family of highly capable multimodal models, arXiv preprint arXiv:2312.11805 (2025). URL: <https://arxiv.org/abs/2312.11805>.
- [25] R. van der Goot, A. Üstün, A. Ramponi, I. Sharaf, B. Plank, Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP, in: D. Gkatzia, D. Seddah (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2021, pp. 176–197. URL: <https://aclanthology.org/2021.eacl-demos.22/>. doi:10.18653/v1/2021.eacl-demos.22.
- [26] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv, C. Zheng, D. Liu, F. Zhou, F. Huang, F. Hu, H. Ge, H. Wei, H. Lin, J. Tang, J. Yang, et al., Qwen3 technical report, arXiv preprint arXiv:2505.09388 (2025). URL: <https://arxiv.org/abs/2505.09388>.
- [27] A. Ramponi, B. Plank, Neural unsupervised domain adaptation in NLP—A survey, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 6838–6855. URL: <https://aclanthology.org/2020.coling-main.603/>. doi:10.18653/v1/2020.coling-main.603.
- [28] C. Casula, S. Vecellio Salto, A. Ramponi, S. Tonelli, Delving into qualitative implications of synthetic data for hate speech detection, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 19709–19726. URL: <https://aclanthology.org/2024.emnlp-main.1099/>. doi:10.18653/v1/2024.emnlp-main.1099.
- [29] T. Wullich, A. Adler, E. Minkov, Fight fire with fire: Fine-tuning hate detectors using large samples of generated hate speech, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2021, Association for Computational Linguistics, Punta Cana, Dominican Republic, 2021, pp. 4699–4705. URL: <https://aclanthology.org/2021.findings-emnlp.402/>. doi:10.18653/v1/2021.findings-emnlp.402.
- [30] J. Tiedemann, S. Thottingal, OPUS-MT – building open translation services for the world, in: A. Martins, H. Moniz, S. Fumega, B. Martins, F. Batista, L. Coheur, C. Parra, I. Trancoso, M. Turchi, A. Bisazza, J. Moorkens, A. Guerbero, M. Nurminen, L. Marg, M. L. Forcada (Eds.), Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, European Association for Machine Translation, Lisboa, Portugal, 2020, pp. 479–480. URL: <https://aclanthology.org/2020.eamt-1.61/>.
- [31] A. Daffara, A. Ramponi, S. Tonelli, WorthIt: Check-worthiness estimation of Italian social media posts, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 337–351. URL: <https://aclanthology.org/2025.clicit-1.35/>.
- [32] P. Goffredo, M. Chaves, S. Villata, E. Cabrio, Argument-based detection and classification of fallacies in political debates, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 11101–11112. URL: <https://aclanthology.org/2023.emnlp-main.684/>. doi:10.18653/v1/2023.emnlp-main.684.

Label at FadeIT: Fallacy-Aware LLM Reasoning for Score-Based Classification

Tiziano Labruna¹, Eleni Papadopoulos^{1,2}

¹Università degli Studi di Padova, Italy

²Politecnico di Torino, Italy

Abstract

Recognizing fallacies in everyday argumentation is a crucial step towards fostering critical thinking and mitigating the spread of harmful discourse, especially in online environments. In this paper, we present our submission to Subtask A of the FadeIT shared task at EVALITA 2026, which focuses on coarse-grained fallacy detection in Italian social media posts. Our approach leverages large language models (LLMs) as structured reasoning components rather than end-to-end classifiers. Inspired by strategy-aware prompting methods, we decompose the detection process by independently assessing each fallacy type through a two-step prompting procedure: first eliciting a targeted analysis, and then producing a graded score indicating the strength of the fallacy's presence. These scores are subsequently employed in two distinct configurations: a threshold-based approach that directly converts scores into predictions using optimized cutoffs, and a hybrid architecture that combines scores with a supervised multi-label classifier trained to model each annotator's judgments independently. This method aims to combine the interpretative capabilities of LLMs with the robustness of supervised learning, while explicitly accounting for annotator disagreement.

Keywords

Fallacy Detection, Argumentation Mining, Large Language Models, Strategy-aware Prompting, Multi-label Classification, Annotator Disagreement

1. Introduction

Fallacies are widespread in everyday argumentation, particularly in online public discourse, where complex social and political issues are often discussed in emotionally charged and polarized contexts. This scenario justifies the necessity of recognizing fallacious reasoning in order to promote fair debate and limit the spread of disinformation. Recognizing this need, a growing number of researchers in natural language processing have focused on methods to automatically detect fallacies in text, spanning different domains, languages, and levels of granularity.

Early work on fallacy detection framed the task as a supervised problem, adopting traditional machine learning methods and neural models [1, 2] combined with hand-crafted lexical and contextual features. More recent approaches leverage pre-trained language models in an end-to-end fashion [3, 4, 5, 6, 7, 8] to better capture the subtle and context-dependent nature of fallacious reasoning. However, fallacy detection remains a particularly challenging task due to several factors: the abstract and overlapping nature of fallacy definitions, the frequent co-occurrence of multiple fallacies within the same text, and the presence of genuine disagreement among human annotators when interpreting argumentative content, a scenario of high uncertainty where model judgments can become unstable [9].

The FadeIT shared task [10] addresses these challenges by explicitly embracing annotator disagreement rather than collapsing it into a single ground truth, through the FAINA dataset [11] that models human label variation. In Subtask A, participants are required to perform coarse-grained fallacy detection, identifying which of 20 predefined fallacy types are present in a given post. Importantly, systems are evaluated against multiple gold standards corresponding to different annotators, reflecting the subjective nature of the task.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ tiziano.labruna@unipd.it (T. Labruna); eleni.papadopoulos@polito.it (E. Papadopoulos)

ORCID 0000-0001-7713-7679 (T. Labruna); 0009-0002-0994-5142 (E. Papadopoulos)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

In this work, we propose an approach that departs from purely using LLMs as end-to-end classifiers for fallacy detection. Instead, drawing inspiration from research on teaching LLMs to make strategic decisions about resource use [12] and acknowledging the limitations of their consistency in constrained tasks [13, 14], we leverage LLMs as intermediate reasoning agents that provide structured, fallacy-specific assessments of a text. Our method is inspired by recent work on strategy-aware prompting, such as PCoT [15], which injects knowledge about persuasive strategies into the reasoning process of LLMs. While PCoT focuses on jointly modeling multiple strategies in the context of disinformation detection, our approach adapts this idea to fallacy detection by considering each fallacy type independently.

Our method proceeds in two steps for each fallacy type: first, the LLM generates a short analytical explanation discussing potential presence of a specific fallacy in the text; second, it produces a numerical score from 1 to 10 reflecting the degree of fallacy presence, based on both the original text and its own analysis. This structured, chain-of-thought process is designed to enhance reasoning stability, a known challenge for LLMs in ambiguous tasks [16]. The resulting fallacy-specific scores are then used in two ways: first, scores serve as features for a supervised multi-label classifier that, rather than aggregating annotations, learns to model each annotator’s judgments independently. In this way, the model aims to capture not only the presence of fallacies, but also systematic differences in how annotators interpret and apply fallacy definitions. In the second approach, we experiment with empirically selecting the optimal threshold for each score to improve prediction accuracy.

Our contributions can be summarized as follows: (I) We introduce a fallacy-aware prompting framework that decomposes coarse-grained fallacy detection into fallacy-specific reasoning steps, leveraging large language models to produce structured analyses and graded assessments for each fallacy type; (II) We assess the LLM-generated scores through an empirical threshold-selection strategy to optimize prediction performance for each fallacy; (III) We propose a hybrid architecture that combines LLM-generated fallacy scores with a supervised multi-label classifier, enabling robust prediction while avoiding fully end-to-end reliance on LLMs; (IV) We explicitly model annotator disagreement by training the classifier to predict the labels of each annotator separately, rather than collapsing annotations into a single gold standard.

2. Data and Resources

2.1. Dataset

Our experiments are conducted on the data released for the FadeIT [10] shared task as part of the EVALITA 2026 evaluation campaign. The dataset, provided by the task organizers, is based on FAIRNA [11] and consists of Italian social media posts discussing migration, climate change, and public health over a four-year time span. Each post is annotated for the presence of fallacies belonging to a predefined set of 20 fallacy types.

For Subtask A (coarse-grained fallacy detection), annotations are provided at span level by two independent annotators and then aggregated for each post. Each annotator may assign zero, one, or multiple fallacy labels to a post, reflecting the multi-label nature of the task. Importantly, the dataset preserves individual annotator labels rather than aggregating them into a single gold standard, in line with recent work highlighting the importance of modeling human label variation in subjective NLP tasks [17].

The data are distributed in a tab-separated format including post identifiers, temporal metadata, topic keywords, anonymized post text, and separate columns for each annotator’s labels. The organizers provide a train/development split, corresponding to 80% of the data, and a held-out test set comprising the remaining 20%, which is used for the official evaluation. In our approach, the LLM-based analysis pipeline takes as input only the post identifier, the post text, and, in some settings, the topic while annotator-specific labels are employed to train the supervised classifier.

2.2. Language Model and Computational Resources

The proposed approach relies on a large language model as a reasoning component for fallacy-specific analysis and scoring. All experiments are performed using GEMMA-3-12B, an instruction-tuned multilingual open large language model released by Google. The model is accessed through the Hugging Face ecosystem using the official `transformers` library and is employed in inference-only mode, without any form of fine-tuning or parameter updates.

Model inference is carried out via API-based loading of the pre-trained weights (`google/gemma-3-12b-it`), using automatic device placement on a single NVIDIA A40 GPU. The model is executed in `bf16` precision to balance computational efficiency and numerical stability, and attention is computed using an eager implementation compatible with the employed PyTorch version. A corresponding processor is used for efficient tokenization and input preparation.

By relying exclusively on prompting and inference-time reasoning, our setup avoids task-specific fine-tuning and enables a clear separation between the LLM-based reasoning stage and the downstream supervised classifier. In practice, the generation stage requires $n \times k$ analyses, where n is the number of posts and k is the number of fallacy classes, since each post is evaluated independently for every fallacy type. This design choice allows the model to be treated as a reusable and modular component, while keeping computational requirements within the constraints typically available to shared task participants.

3. Methodology

Our approach leverages LLM reasoning to produce fallacy-specific scores that serve as the foundation for two classification strategies. Inspired by the strategy-aware prompting framework proposed in PCoT [15], we adapt the methodology to the fallacy detection setting by treating each fallacy type as an independent reasoning target. In this section, we outline the hybrid architecture, which consists of: (i) fallacy-specific analysis and scoring using an LLM to generate numerical features, (ii) a supervised multi-label classifier that combines these scores with contextual information. As an alternative, we also evaluate direct threshold-based classification of the LLM scores.

3.1. Overview

Given a social media post, the system processes it independently for each of the 20 fallacy types defined in the FadeIT task. For each fallacy type, the LLM is asked to generate a short analytical paragraph discussing the possible presence of a specific fallacy in the text, with the prompt explicitly including a description of that fallacy type. In the second step, the model is prompted again to produce a numerical score on a fixed scale from 1 to 10, indicating the degree to which the fallacy is present, conditioned both on the original text and on the previously generated analysis. This results in a vector of 20 scalar scores per post. These scores are subsequently used as input features to a supervised multi-layer perceptron (MLP), which predicts fallacy presence for each annotator separately.

This design allows us to decouple fallacy-specific reasoning from the final classification step, while explicitly modeling annotator disagreement.

3.2. Two-step Fallacy-aware Prompting

For each post and each fallacy type, we employ a two-step prompting strategy.

Step 1: Fallacy-specific analysis. In the first step, the LLM is asked to analyze the post with respect to a single fallacy type. The prompt explicitly injects domain knowledge by including a definition of the target fallacy, derived from the official task guidelines. The model is instructed to focus exclusively on the selected fallacy and to be conservative in its judgment, assuming absence when evidence is insufficient.

Formally, given a post p and a fallacy type f_i , the LLM produces an analysis text a_i describing whether and how f_i manifests in p .

The prompt used is presented as follows. All prompts are written in Italian, matching the language of the analyzed texts.

Fallacy-specific system prompt (example).

Sei un assistente che rileva le fallacie nei testi. Le fallacie sono definite come tipi di ragionamento che sembrano validi ma non lo sono. Si verificano quando qualcuno commette un errore nell'argomentazione, sia con l'intento di persuadere il pubblico sia in modo involontario. Queste fallacie sono suddivise in diverse categorie. La tua competenza e il tuo focus sono sull'Ad hominem. Ad hominem: Un attacco personale a un individuo o a un gruppo che devia dalla tesi principale.

You are an assistant that detects fallacies in texts. Fallacies are defined as types of reasoning that appear valid but are not. They occur when someone makes an error in argumentation, either with the intent to persuade the audience or unintentionally. These fallacies are divided into different categories. Your expertise and focus are on Ad hominem. Ad hominem: A personal attack against an individual or a group that diverts from the main claim.

Fallacy-specific user prompt (example).

*Dato un certo testo, valuta criticamente il suo potenziale di fallacia. Inoltre, analizza se il testo utilizza il tipo di fallacia **Ad hominem**. Spiega come l'Ad hominem appare o non appare nel testo. Sii prudente nella valutazione e, se non sei completamente certo che la fallacia sia presente, assumi che non lo sia.*

*Given a certain text, critically evaluate its potential fallaciousness. Additionally, analyze whether the text uses the **Ad hominem** fallacy. Explain how the Ad hominem appears or does not appear in the text. Be cautious in your assessment and, if you are not completely certain that the fallacy is present, assume that it is not.*

Equivalent prompts are defined for all remaining fallacy types, each including a task-specific definition provided by the organizers.

Step 2: Numerical scoring. In the second step, the LLM receives both the original post and the analysis generated in Step 1, and is asked to output a numerical score from 1 to 10 representing the degree of presence of the target fallacy in the post. A higher score indicates stronger or clearer manifestation of the fallacy.

This step enforces a structured output format to facilitate automatic parsing and downstream processing.

The prompts used are presented as follows.

Scoring system prompt.

Sei un Rilevatore di Fallacie. Il tuo obiettivo è valutare il grado di fallacie presenti in un testo. Le fallacie sono definite come tipi di ragionamento che sembrano validi ma non lo sono. Si verificano quando qualcuno commette un errore nell'argomentazione, sia con l'intento di persuadere il pubblico sia in modo involontario. Dovrai dare un punteggio da 1 a 10 sulla base di quanto fallaci sono le argomentazioni nel testo.

You are a Fallacy Detector. Your goal is to assess the degree of fallacious reasoning present in a text. Fallacies are defined as types of reasoning that appear valid but are not. They occur when someone makes an error in argumentation, either with the intent to persuade the audience or unintentionally. You must assign a score from 1 to 10 based on how fallacious the arguments in the text are.

Scoring user prompt.

Dato un testo e un'analisi dei possibili tipi di fallacie, devi rispondere con un numero da 1 a 10 in base al grado di presenza di fallacie nel testo, preceduto da una breve spiegazione del tuo ragionamento. Fornisci la tua risposta sotto forma di dizionario: {"explanation": "...", "response": "..."}.

Given a text and an analysis of the possible types of fallacies, you must respond with a number from 1 to 10 based on the degree to which fallacies are present in the text, preceded by a brief explanation of your reasoning. Provide your answer in dictionary form: {"explanation": "...", "response": "..."}.

Note that the end of each of the fallacy-specific system prompts, the correspondent fallacy definition is directly embedded, as taken from the organizer's guidelines. This ensures that the LLM's reasoning is grounded in the task-specific interpretation of each fallacy and avoids cross-contamination between categories.

The two-step prompting process is repeated independently for all fallacy types, yielding a vector of scores:

$$\mathbf{s} = (s_1, s_2, \dots, s_{20}),$$

where each $s_i \in \{1, \dots, 10\}$ is relative to a fallacy type.

For each post, the final output of the LLM stage is the complete vector of fallacy scores. No aggregation across fallacy types is performed at this stage, as each score is treated as an independent feature capturing the model's confidence in the presence of a specific fallacy. These vectors are stored and reused across runs to ensure reproducibility and to avoid redundant inference calls.

3.3. MLP-based Aggregation of LLM Scores and Human Annotations

To combine the fallacy-related signals extracted from LLMs with human annotations, a supervised multi-layer perceptron (MLP) classifier is employed. The goal of this component is to map a structured representation derived from LLM outputs to fallacy labels, while explicitly modeling annotator-level disagreement.

Each input instance corresponds to a single post and is represented as a feature vector built from three main components:

- **LLM-based fallacy scores.** For each fallacy category, the numeric score produced by the LLM prompting pipeline is included as a feature. The ordering of these scores follows a fixed fallacy list shared across all data splits.
- **Aggregate statistical features.** In addition to raw scores, three summary statistics are computed over the fallacy score vector: mean, variance, and entropy. These features provide a compact characterization of the overall distribution of fallacy scores for a post.
- **Topic information (optional).** When enabled, topic metadata associated with each post is incorporated either as a one-hot encoded vector or as a normalized scalar index. An explicit `__UNKNOWN__` category is used to handle unseen topics.

The resulting feature vector is used as input to an MLP composed of a configurable number of hidden layers, each followed by a ReLU activation function. The final linear layer outputs a vector of size $2 \times N$, where N is the number of fallacy categories. This structure reflects the presence of two independent annotators in the dataset: the first N dimensions correspond to annotator 1, and the remaining N to annotator 2.

Formally, given an input feature vector $\mathbf{x}$, the model computes:

$$\mathbf{z} = f_{\theta}(\mathbf{x}),$$

where f_{θ} denotes the MLP and $\mathbf{z} \in \mathbb{R}^{2N}$ are the output logits. A sigmoid function is applied independently to each logit to obtain per-fallacy probabilities.

Training is performed using binary cross-entropy with logits, allowing each fallacy label for each annotator to be treated as an independent binary classification problem. This design choice enables the

model to learn asymmetric or inconsistent labeling patterns across annotators, rather than forcing a single aggregated gold label.

During inference, probabilities are thresholded to obtain binary predictions. Posts without gold annotations are included in the prediction phase but excluded from metric computation, which is performed using micro-averaged scores over all fallacy-annotator pairs.

3.4. Threshold-based classification

As an alternative to the MLP-based approach, we also evaluated a simpler optimized threshold method that directly converts LLM scores into binary predictions. For each fallacy type and each annotator, a threshold value was selected to maximize the F1 score per-fallacy on the training set. Fallacy classes with scores above the threshold are classified as positive, while those below are classified as negative. The optimized thresholds were then evaluated on the validation set. This approach offers several advantages: it provides greater interpretability, requires no training phase and allows us to assess whether the LLM-generated scores are already well-calibrated for classification without additional supervised learning.

4. Experimental Setup

A series of experiments was conducted to identify the most effective configuration for aggregating LLM-generated fallacy scores and auxiliary information. As described in Section 3, the starting point for all experiments consists of the 20 fallacy scores in the range $[1, 10]$ produced by the LLM, one for each fallacy category.

Different feature configurations were explored by selectively incorporating topic information, raw LLM scores, and aggregate statistical features. In particular:

- **Topic** indicates that topic metadata associated with each post (migration, climate change, public health) was included as an additional feature.
- **Topic (one-hot)** denotes a one-hot encoding of the topic labels.
- **Scores** refers to the inclusion of the 20 raw fallacy scores generated by the LLM.
- **Stats** refers to the inclusion of three aggregate statistics computed over the fallacy score vector, namely average, variance, and entropy.

For each feature configuration, the MLP classifier described in Section 3 was trained using the best hyperparameters selected via grid search. The explored hyperparameter space is reported below:

```
hidden_dim: {32, 64, 128}
hidden_layers: {1, 2, 3}
lr: {1e-3, 5e-4, 1e-4}
batch_size: {8, 16, 32, 64, 128}
```

Evaluation follows the shared task protocol and reports Micro F_1 scores, as well as Micro- F_1 computed separately for each annotator. Results are reported in Table 1.

5. Model Selection and Analysis

The experimental results provide several insights into the behavior of the proposed hybrid approach and its underlying design choices. A first observation concerns the overall effectiveness of incorporating topic information. As shown in Table 1, configurations that include topic features consistently outperform those relying solely on LLM-generated fallacy scores and their aggregate statistics. In particular, using topic information in combination with scores and stats yields the highest Micro F_1 scores, as well as the best performance for both annotators. This suggests that topical context plays a central role in

Table 1

Results on the FAINA validation set for different feature configurations. Ann. 1 and Ann. 2 report Micro F_1 computed separately for each annotator. Green intensity indicates higher values with proportions computed for each column.

Configuration	Micro F_1	Ann. 1	Ann. 2
Scores	39.01	30.14	42.21
Scores + Topic	40.09	30.22	47.60
Scores + Topic (one-hot)	39.18	33.42	43.82
Stats	38.23	28.77	44.98
Stats + Topic	38.69	30.05	45.61
Stats + Topic (one-hot)	40.05	28.98	47.89
Scores + Stats	37.59	28.80	44.84
Scores + Stats + Topic	43.67	36.68	49.38
Scores + Stats + Topic (one-hot)	42.32	35.20	48.05
Threshold-based	39.92	36.91	48.79
Baseline	20.84	15.84	25.84

fallacy detection, possibly by constraining the plausible set of fallacies and reducing ambiguity at the classification stage.

In contrast, the combination of both LLM scores and aggregate statistics does not systematically improve performance compared to those features taken individually, likely because it results to be redundant or noisy. In particular, the aggregate statistics are directly derived from the same score distributions and therefore provide limited additional information beyond what the classifier can already infer from the raw scores themselves.

The threshold-based approach demonstrates surprisingly competitive performance despite its simplicity, achieving a Micro- F_1 of 39.92%. This indicates that the LLM-generated scores are reasonably well-calibrated for direct classification without necessarily requiring additional supervised learning, though combining them with scores statistics and topic information still yields the best performance.

Another notable aspect emerges from the annotator-level analysis. Across all configurations, a consistent performance gap is observed between Annotator 1 and Annotator 2, with the latter achieving higher Micro- F_1 scores. This suggests the presence of substantial annotator disagreement in the dataset and supports the decision to model annotators separately rather than collapsing their labels into a single gold standard. The results indicate that the proposed MLP is able to capture annotator-specific labeling patterns, even when overall performance remains moderate.

From a broader perspective, these findings shed light on the theoretical motivation behind the adopted approach. By decomposing the fallacy detection task into fallacy-specific reasoning steps and isolating each category during the LLM prompting phase, the model explicitly mirrors the way fallacies are defined and annotated in the dataset, thus the experimental results support the theoretical intuition that fallacy detection benefits from category-wise isolation and explicit reasoning. We submitted our three best-performing configurations from the development set to the test phase.

6. Results

Results (Table 2) show that Run 3 (threshold-based) achieves the highest recall among all submitted runs (68.08% micro-averaged and 60.13% macro-averaged), though at the cost of precision. Despite this different optimization strategy, all three runs achieve similar overall F_1 scores (38-39% Micro F_1 , 31-33% Macro F_1), though Runs 2 and 3 exhibit limited prediction diversity, concentrating on a subset of fallacy types.

Our systems generally demonstrate the strongest performance on fallacies such as *Loaded Language* (64.40% F_1 score on average), *Appeal to Emotion* (61.12% on average), *Vagueness* (52.01% on average), and *Name Calling* (58.47% on Run 3). Analysis of annotator-specific performance reveals systematic disagreement patterns across fallacy types, with Annotator 2 consistently outperforming Annotator 1

Table 2

Results on the FAINA test set. Annotator columns (Ann. 1, Ann. 2) report micro-F₁ computed separately for each annotator. **Bold**: best score within each metric column. Bold: best score across all submissions in the task. Green intensity indicates higher values with proportions computed for each column.

Run Configuration		Micro			Macro			Annotator	
		P	R	F ₁	P	R	F ₁	Ann. 1	Ann. 2
1	Score+Stats+Topic	52.82	30.94	38.96	41.52	26.99	30.94	34.26	43.65
2	Score+Stats+Topic (one-hot)	52.76	30.11	38.32	43.01	26.81	30.84	33.98	42.66
3	Threshold-based	27.60	68.08	39.26	24.64	60.13	33.37	37.39	41.13
	Baseline	38.53	14.28	20.84	15.13	3.45	5.10	15.84	28.85

across all configurations by 3.74-9.39 F₁ points, with the smallest gap observed in Run 3. In particular, the class *Vagueness* exhibits the largest systematic difference with Annotator 2 outperforming Annotator 1 by 27.56% across all runs.

7. Conclusions

This paper presented a two-stage approach to coarse-grained fallacy detection for Italian social media posts, developed in the context of the FadeIT shared task at EVALITA 2026. Our method leverages large language models as structured reasoning components through a two-step prompting procedure that decomposes the detection task into independent evaluations for each of the 20 fallacy types, generating numerical scores that capture both the presence and strength of individual fallacies. These LLM-generated scores are exploited in two complementary ways: (i) a threshold-based approach that applies empirically optimized cutoffs directly to the scores for classification and (ii) a hybrid architecture that combines the scores with a supervised multi-label MLP classifier, optionally augmented with topic and statistical features. In the hybrid configuration, the classifier models each annotator’s judgments independently, aiming to capture systematic differences in how annotators interpret and apply fallacy definitions.

Preliminary experimental results for model selection on Subtask A demonstrate that both approaches achieve competitive performance. On the validation set, the hybrid configuration incorporating scores, aggregated statistics and topic information achieved the highest performance (43.67% Micro F₁), while the threshold-based approach proved remarkably effective given its simplicity (39.92% Micro F₁). In the test phase, the threshold-based approach attained the highest recall among all submitted systems (68.08% micro-averaged, 60.13% macro-averaged). Despite different optimization strategies, all three submitted runs achieved similar overall F₁ scores (38-39% micro F₁, 31-33% macro F₁).

Beyond task-specific performance, this work contributes to the broader discussion on how LLM-based reasoning can be integrated into supervised learning pipelines for subjective NLP tasks. By processing each fallacy type independently through dedicated prompts, our approach enhances transparency: each classification decision can be traced back to a specific analytical explanation and numerical score for that fallacy. At the same time, more sophisticated architectures aimed at transferring reasoning into downstream classifiers could potentially improve classification accuracy.

Future work may explore alternative representations of LLM outputs, such as structured explanations or uncertainty-aware features, as well as extensions to fine-grained span-level detection. More generally, this line of research points toward hybrid systems that balance interpretability, contextual awareness, and robustness to human disagreement in fallacy detection and related argumentation tasks.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5 and Gemini-3 in order to conduct grammar and spelling check. After using these tools, the authors reviewed and edited the content as needed and

take full responsibility for the publication’s content.

References

- [1] I. Habernal, P. Pauli, I. Gurevych, Adapting serious game for fallacious argumentation to German: Pitfalls, insights, and best practices, in: N. Calzolari, K. Choukri, C. Cieri, T. Declerck, S. Goggi, K. Hasida, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, S. Piperidis, T. Tokunaga (Eds.), *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, European Language Resources Association (ELRA), Miyazaki, Japan, 2018. URL: <https://aclanthology.org/L18-1526/>.
- [2] I. Habernal, H. Wachsmuth, I. Gurevych, B. Stein, Before name-calling: Dynamics and triggers of ad hominem fallacies in web argumentation, in: M. Walker, H. Ji, A. Stent (Eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 386–396. URL: <https://aclanthology.org/N18-1036/>. doi:10.18653/v1/N18-1036.
- [3] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7180–7198. URL: <https://aclanthology.org/2022.findings-emnlp.532/>. doi:10.18653/v1/2022.findings-emnlp.532.
- [4] M. Chaves, E. Cabrio, S. Villata, Falcon: A multi-label graph-based dataset for fallacy classification in the covid-19 infodemic, in: *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing, SAC ’25*, Association for Computing Machinery, New York, NY, USA, 2025, p. 988–995. URL: <https://doi.org/10.1145/3672608.3707913>. doi:10.1145/3672608.3707913.
- [5] P. Goffredo, M. Chaves, S. Villata, E. Cabrio, Argument-based detection and classification of fallacies in political debates, in: H. Bouamor, J. Pino, K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, 2023, pp. 11101–11112. URL: <https://aclanthology.org/2023.emnlp-main.684/>. doi:10.18653/v1/2023.emnlp-main.684.
- [6] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, 2023. URL: <https://arxiv.org/abs/2301.09992>. arXiv:2301.09992.
- [7] Y. Lei, R. Huang, Boosting logical fallacy reasoning in llms via logical structure tree, 2024. URL: <https://arxiv.org/abs/2410.12048>. arXiv:2410.12048.
- [8] J. Jeong, H. Jang, H. Park, Large language models are better logical fallacy reasoners with counter-argument, explanation, and goal-aware prompt formulation, 2025. URL: <https://arxiv.org/abs/2503.23363>. arXiv:2503.23363.
- [9] T. Labruna, S. Gallo, G. Da San Martino, Positional bias in binary question answering: How uncertainty shapes model preferences, in: *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025)*, 2025, pp. 550–560.
- [10] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [11] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.

- [12] T. Labruna, J. A. Campos, G. Azkune, H. Center-Ixa, When to retrieve: Teaching llms to utilize information retrieval effectively, *Context* 12 (2024) 14.
- [13] T. Labruna, S. Brenna, G. Bonetta, B. Magnini, Are you a good assistant? assessing llm trustability in task-oriented dialogues, in: *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, 2024, pp. 470–477.
- [14] T. Labruna, B. Magnini, Evaluating task-oriented dialogue consistency through constraint satisfaction, *arXiv preprint arXiv:2407.11857* (2024).
- [15] A. Modzelewski, W. Sosnowski, T. Labruna, A. Wierzbicki, G. Da San Martino, Pcot: Persuasion-augmented chain of thought for detecting fake news and social media disinformation, in: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025, pp. 24959–24983.
- [16] S. Casola, T. Labruna, A. Lavelli, B. Magnini, Testing chatgpt for stability and reasoning: A case study using italian medical specialty tests, in: *Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2023)*, 2023.
- [17] B. Plank, The “problem” of human label variation: On ground truth in data, modeling and evaluation, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 10671–10682. URL: <https://aclanthology.org/2022.emnlp-main.731/>. doi:10.18653/v1/2022.emnlp-main.731.

MALTO at FadeIT: A BERT-Based System for Multi-Label Fallacy Detection in Italian Social Media

Matin Salami¹, Luca M. Rodia¹, Vladimir Schiau¹, Evren A. Munis¹, Claudio Savelli¹ and Flavio Giobergia¹

¹ Politecnico di Torino, Turin, Italy

Abstract

Fallacies in social media posts are common and can make texts less trustworthy, and more likely to foster biased views. In this work, we study multi-label fallacy classification in Italian social media posts by detecting 20 different fallacy types that can appear in the same post. We address this task using encoder-based language models. In particular, we build a fallacy classifier based on ALBERTo, a BERT-style encoder for Italian, and use a global multi-label threshold to turn the model's outputs into the final set of fallacy labels. Our results indicate that paraphrase-based augmentation can degrade performance in fallacy detection, likely by introducing label noise, while the choice of loss function substantially affects the trade-off between micro- and macro-averaged performance.

Keywords

fallacy detection, multi label classification, encoder based models, deep learning, natural language processing

1. Introduction

Fallacious reasoning is pervasive in online public discourse, where emotionally charged and fast-paced interactions often foster misleading argumentative strategies [1, 2]. Detecting such fallacies is crucial for supporting critical thinking, promoting healthier public debate, and mitigating the spread of manipulative or harmful content [3]. However, this task is particularly challenging due to the subjective nature of the interpretation of the fallacy and the resulting disagreement among human annotators.

The FadeIT [4] (Fallacy Detection in Italian Social Media Texts) shared task, proposed within the EVALITA 2026 [5] campaign, addresses this problem for the Italian language by focusing on fallacy detection in social media posts over a four-year time span [4]. This paper focuses on Subtask A, which targets coarse-grained, post-level fallacy detection. The task is formulated as a multi-label classification problem, where each post may contain zero, one, or multiple fallacies, as annotated by two annotators. This setting reflects real-world conditions, where argumentative interpretation is often ambiguous rather than governed by a single ground truth.

In this work¹, we propose a classifier based on encoder-only transformer-based models. In the submitted results, we used the ALBERTo model [6], which encodes each social media post into token-level representations. These embeddings are aggregated via mean pooling to obtain a fixed-size representation of the post, which is then fed to a lightweight classification head for multi-label prediction.

Our best approach, described above, achieves a micro-averaged F1 score of 0.54 on the official test set for Subtask A. Using different configurations with augmented data and weighted loss resulted in a score of 0.374, while an additional approach incorporating description awareness reached 0.466 on the test set.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT
✉ s306400@studenti.polito.it (M. Salami); s338250@studenti.polito.it (L. M. Rodia); s353706@studenti.polito.it (V. Schiau); evrenayberk.munis@studenti.polito.it (E. A. Munis); claudio.savelli@polito.it (C. Savelli); flavio.giobergia@polito.it (F. Giobergia)

✉ 0000-0002-0877-7063 (C. Savelli); 0000-0001-8806-7979 (F. Giobergia)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹The code to replicate the experiments can be found at <https://github.com/MAL-TO/FADEIT-2026>

2. Related Works

This challenge addresses two main obstacles: handling an unbalanced dataset and extracting meaningful information from unstructured text. The following subsections review the main approaches found in the literature to solve these problems

Multilabel Classification: A method proposed by Lanchantin et al. [7] adapts transformer architectures for multilabel classification. In this framework, the image features are processed alongside label embeddings, which are randomly masked during training to predict the full label set. This strategy effectively captures label correlations while maintaining robustness at inference time; by training on instances where all labels are masked, the model also learns to perform accurate classification solely from visual data when no partial label information is available.

A sequence-to-sequence approach is proposed in Yang et al. [8], where the authors apply an attention mechanism to adaptively select the most informative words for each predicted label. This model treats multi-label classification as a generation problem, capturing high-order correlations by conditioning the prediction of the next label on the previously generated ones. Furthermore, they introduce a Global Embedding mechanism: this combines the embedding of the top-predicted label with a probability-weighted average of the entire label distribution from the previous time step.

Finally, Da San Martino et al. [9] examines propaganda techniques and logical fallacies in news articles, and proposes a Multi-Granularity Network. In this architecture, a sentence-level classification drives a 'Gate' mechanism: if a sentence is predicted as non-propagandistic, the gate suppresses further fragment-level predictions, thereby significantly reducing false positives.

Multilabel Classification under Class Imbalance: A major challenge in fallacy classification is the highly skewed distribution of labels. For a better description go to [data distribution](#). Existing approaches mainly address this problem through loss re-weighting strategies and data-level balancing techniques. Loss-based methods adapt the standard cross-entropy to weight samples differently based on their ground-truth labels, aiming to reduce the influence of frequent classes. Focal Loss [10] achieves this by down-weighting easy examples and emphasizing harder instances, effectively focusing on the model's certainty. Similarly, Class-Balanced Loss [11] introduces the notion of effective information, observing that the marginal information provided by each additional example decreases as class frequency increases. The loss is reduced proportionally to the class frequency, allowing the model to focus on underrepresented samples.

Data Augmentation via LLM An alternative strategy for addressing long-tailed label distributions is data augmentation via oversampling, which artificially rebalances the dataset by increasing the number of instances for minority classes [12]. A representative method in this line of research is that of He et al. [13], in which the authors generate synthetic samples in proportion to a balancing coefficient $\beta \in [0, 1]$. In this framework, $\beta = 1$ corresponds to a fully balanced dataset, while $\beta = 0$ reduces to the non-augmented data distribution. When the augmented data are used to train a decision tree classifier, the authors report consistent improvements over both the non-augmented setting and the SMOTE approach [14], achieving approximate recall gains of +8% and +32% respectively. More recent approaches to data augmentation for logical fallacy detection rely on large language models (LLMs) to generate synthetic argumentative content. This direction is explored by Poliakov and Shvai [15], who employ a Retrieval-Augmented Generation (RAG) framework to condition an LLM on scientific literature, enabling the generation of contextually grounded, plausible, and fallacious arguments. A related approach is presented in Mouchel et al. [16], where GPT-3.5-Turbo is used to generate multiple logical arguments from the EXPLAGRAPHS dataset [17]. Furthermore, Glazkova and Zakharova [18] demonstrate that prompt-based LLM augmentation strategies can yield greater performance compared to models trained only on original data. Following this line of work, we tested ChatGPT to generate paraphrases of selected training samples, which are then added to the training data before model optimization. The paraphrases are generated exclusively from the training set, and the original labels

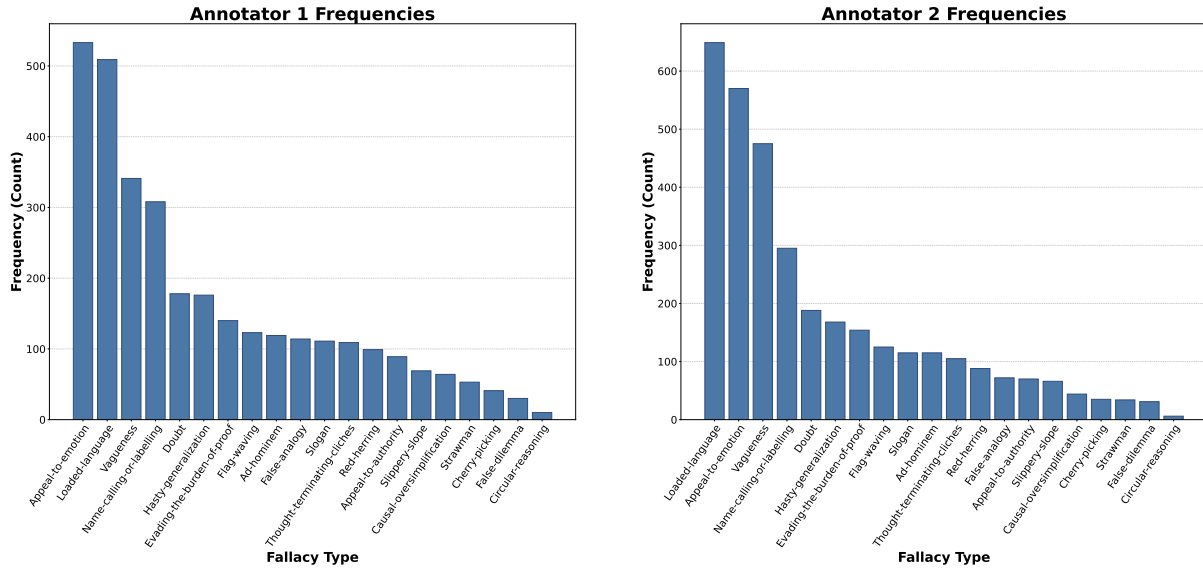


Figure 1: Label counts per annotator across the complete dataset.

are retained for all augmented instances. After augmentation, the dataset remains heavily imbalanced, so we also address this by using different weighted loss functions. The exact prompt used for augmentation is reported in appendix B for reproducibility.

Fallacy detection: A central challenge in text classification is extracting meaningful semantic representations that capture the relevant concepts expressed in the text. This is further amplified in tasks such as logical fallacy detection, where labels often correspond to abstract rhetorical patterns rather than surface-level lexical clues. An effective approach is proposed in Wang et al. [19], where both the input text and the label descriptions are encoded into dense vector representations. Each text and label is mapped to a single embedding, and the resulting representations are subsequently aggregated and fed into a neural classifier. In contrast, Çöltekin et al. [20] explores a more traditional machine learning approach based on Support Vector Machines (SVM) trained on word n-gram features. Despite its relative simplicity, this method remains competitive due to its computational efficiency and interpretability. Moreover, the reduced training cost enables extensive hyperparameter tuning, resulting in reasonable performance even without deep contextual representations.

3. Challenge Description

Task: The objective of subtask A is to perform multi-label classification at the post level. This means that, given a social media post, the model must predict, for both annotators, the set of fallacy types present in the text. The model performance is evaluated using the micro-averaged F1 score, which is the metric used for the main evaluation and submission ranking.

Data: The FadeIT shared task is based on FAINA [4], a dataset for fallacy detection in Italian social media posts. The data, spanning a four-year time period, cover public discourse on three socially relevant topics: migration, climate change, and public health. Posts are anonymized by replacing user mentions and sensitive information with placeholders such as [USER], [URL], [EMAIL], and [PHONE]. The dataset includes 20 fallacy types and provides annotations from two independent annotators, allowing multiple fallacy labels per post and capturing genuine disagreement in fallacy interpretation. The data are split into a train set and a test set. The train set consists of 1152 posts, while the test set contains 288 posts.

Data distribution: Several fallacies represent only a small fraction of the dataset, such as circular reasoning and false dilemma, which combined account for less than 1% resulting in a long-tailed label

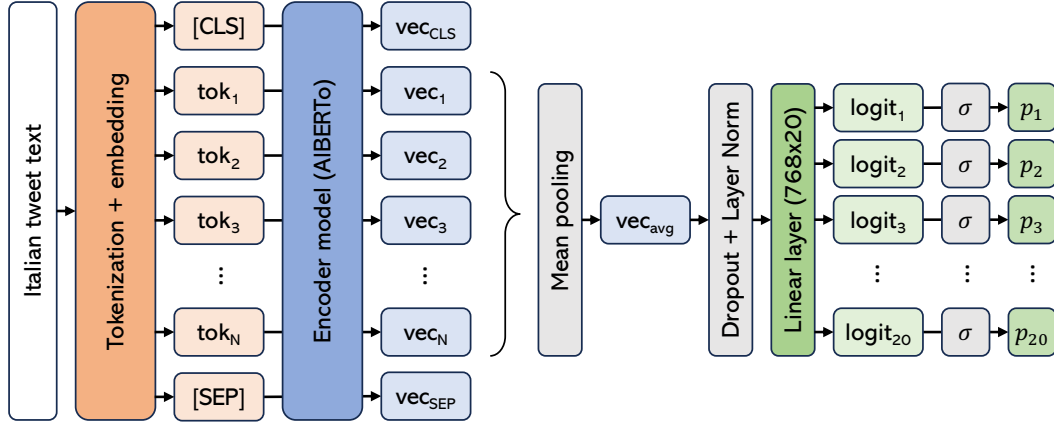


Figure 2: Architecture of the proposed multi-label text classification model. An Italian tweet is tokenized and encoded using AlBERTo. Contextualized token representations are aggregated via attention-mask-aware mean pooling, followed by dropout, layer normalization, and a fully connected classification layer that produces one logit per label. Sigmoid activations are applied during inference to obtain label probabilities.

distribution. This issue is exacerbated in a multilabel setting, where minority labels often co-occur with more frequent ones.

4. Methodology

Our approach is inspired by previous methods for multi-label text classification introduced by Fallah et al. [21]. It builds upon: 1) a BERT-adapted model and 2) a thresholding technique. The two main parts of it are explained in detail in the next two sections.

4.1. BERT Adaptation

Transformer-based pretrained language models such as BERT and its variants have become a dominant approach and have pushed the state of the art across many NLP tasks [22, 23]. Since our dataset consists of Italian tweets from social media, we employ AlBERTo, a BERT-based language model pretrained on approximately 200 million Italian tweets [6]. We further include UmBERTo, another monolingual Italian RoBERTa-based model [24], and XLM-RoBERTa-base, a multilingual transformer [25], as evaluated architectures for the proposed pipeline.

Each text instance in the dataset is annotated according to two distinct professional annotators [26]. Rather than learning separate predictors for the two schemes, we merge the two label sets into a unified label space and train a single multi-label classifier that predicts one shared set of labels.

In our approach, instead of relying on the special [CLS] token, the encoder produces a sequence of contextualized token representations $H = \{h_1, h_2, \dots, h_T\}$ for an input sequence of length T , where each $h_t \in \mathbb{R}^d$ and d denotes the hidden size of the encoder. To obtain a fixed-size representation of the entire input text, we apply mean pooling over the token dimension:

$$h = \frac{1}{T} \sum_{t=1}^T h_t$$

Mean pooling is used to obtain a document-level representation, as global pooling methods (e.g., average pooling) have been found to be particularly effective for classification tasks [27].

The resulting document representation is passed through a dropout layer for regularization and a layer normalization step [28] to improve training stability. Finally, a fully connected linear layer maps the normalized representation to a vector of K output logits, one for each label. The model architecture is depicted in Figure 2.

4.2. Global Thresholding Strategy

Since the task is formulated as multi-label classification, the model outputs are converted into binary predictions using a decision threshold. Instead of fixing this threshold a priori, we adopt a global threshold selection strategy. Specifically, a global classification threshold $\tau \in [0, 1]$ is selected by maximizing the micro-averaged F1 score on the validation set. The threshold is varied from 0 to 1 with a step size of 10^{-1} , and the value yielding the best validation performance is selected. The optimal threshold is then applied to the test set for inference which in our case is 0.5.

4.3. Label Description-Aware Architecture

In addition to the standard setup, we explored a label-aware variant of the BERT-based classifier aimed at explicitly incorporating semantic information from the textual descriptions of fallacy labels. The main motivation was to move beyond a purely document-level classification paradigm and instead model fallacy detection as a semantic compatibility problem between the input text and the conceptual description of each label. In this architecture, each post is encoded using a pretrained Italian BERT model, and the [CLS] token representation is used as the document embedding. Each fallacy label is represented by a learnable embedding initialized from its textual description. For each (text, label) pair, the projected document embedding is concatenated with the corresponding label embedding and fed into a shared feed-forward network that outputs a single binary logit. This process is repeated independently for all labels, producing multi-label predictions for each post. The model is trained using Focal Loss to address class imbalance, and class-specific decision thresholds are selected on the validation set to optimize F1-score. This formulation allows the model to explicitly encode relationships between textual content and fallacy definitions, encouraging semantic alignment.

5. Experimental Setup

This section describes the experimental protocol adopted to evaluate the proposed model and the baseline methods. For all baseline techniques, we maintain the original model implementations to ensure a fair comparison.

5.1. Training Procedure

All models are optimized using AdamW [29] with a learning rate of 2×10^{-5} and trained for 10 epochs. We employ a cosine learning rate schedule with linear warm-up, where the warm-up phase corresponds to the first 10% of the total training steps. Performance is evaluated using micro-averaged precision, recall, and F1-score. All experiments are conducted on a machine equipped with an AMD Ryzen 9 7940HX CPU and an NVIDIA GeForce RTX 4070 GPU. The full configuration table is in appendix A.

To obtain training, validation, and test splits that preserve label imbalance and label co-occurrence patterns, we adopt an iterative stratification strategy. This procedure mitigates the risk of rare labels being absent from the validation or test sets due to random splitting. We partition the data into three subsets: a training set containing 70% of the instances, and validation and test sets each containing 15% of the data.

5.2. Loss Function Configurations

To address class imbalance, we experimented with three loss functions: (i) Binary Cross-Entropy, (ii) Binary Cross-Entropy with positive class weighting, and (iii) Focal Loss.

Binary Cross-Entropy. BCE is a standard choice for multi-label classification, treating each label as an independent binary prediction. This formulation assumes a balanced label distribution and assigns equal importance to all classes, which can be suboptimal in highly imbalanced settings.

Binary Cross-Entropy with Positive Class Weights. To mitigate label imbalance, we employed a weighted variant of BCE that assigns higher importance to positive instances of rare labels. For

Table 1

Performance comparison across experimental settings. **Test set** indicates the evaluation split: *Ours* refers to the internally constructed test set, while *Official* refers to the competition’s official test set.

Method	Test set	Micro-average			Macro-average		
		Precision	Recall	F1	Precision	Recall	F1
<i>Baseline</i>	Official	0.385	0.142	0.208	0.151	0.034	0.051
ALBERTo (BCE with O-data)	Official	0.557	0.536	0.546	0.419	0.309	0.326
ALBERTo (BCE(weights) with Aug-data)	Official	0.374	0.45	0.408	0.235	0.30	0.256
ALBERTo (description-aware with Aug-data)	Official	0.466	0.437	0.450	0.281	0.233	0.232
ALBERTo (BCE with O-data)	Ours	0.574	0.576	0.575	0.36	0.315	0.312
ALBERTo (BCE(weights) with O-data)	Ours	0.436	0.622	0.513	0.318	0.4652	0.37
ALBERTo (Focal with O-data)	Ours	0.621	0.463	0.53	0.349	0.247	0.276
ALBERTo (BCE with Aug-data)	Ours	0.45	0.495	0.471	0.25	0.244	0.2316
ALBERTo (BCE(weights) with Aug-data)	Ours	0.38	0.40	0.43	0.245	0.323	0.27
ALBERTo (Focal loss with Aug-data)	Ours	0.38	0.493	0.43	0.245	0.323	0.27
UmBERTo (BCE with O-data)	Ours	0.5964	0.5539	0.574	0.298	0.273	0.269
XLM-Roberta-Base (BCE with O-data)	Ours	0.613	0.528	0.567	0.288	0.263	0.263

each label i , a positive weight w_i is computed as the ratio between negative and positive samples, $w_i = N_i^- / (N_i^+ + \epsilon)$, where N_i^+ and N_i^- denote the number of positive and negative occurrences of label i , respectively. This strategy increases the contribution of rare labels during training and has been shown to improve robustness in imbalanced multi-label classification settings [30]. In our experiments, the weights were computed directly from the training data.

Focal Loss. The metric extends BCE by down-weighting well-classified examples and focusing training on harder instances. Starting from the per-label BCE loss ℓ_i^{BCE} , we define the probability assigned to the ground-truth class as $p_{t,i} = \exp(-\ell_i^{\text{BCE}})$. The focal loss then applies a modulating factor $(1 - p_{t,i})^\gamma$ and an α -balancing term to reweight positive and negative targets, yielding:

$$\mathcal{L}_{\text{FL}} = \sum_{i=1}^C \alpha_{t,i} (1 - p_{t,i})^\gamma \ell_i^{\text{BCE}} \quad (1)$$

where $\gamma \geq 0$ controls the focus on hard examples. This objective was originally introduced for dense object detection [10] and has since been adopted in imbalanced multi-label learning.

5.3. Data Augmentation with ChatGPT

Following prior work on LLM-based data augmentation, we leveraged ChatGPT to generate paraphrases of selected training instances, which were subsequently added to the training set prior to model optimization. Importantly, paraphrases were generated exclusively from the training data to prevent data leakage, and the original labels were preserved for all augmented samples. To specifically target class imbalance, we generated two paraphrases for each instance belonging to fallacy labels with fewer than 100 training examples. This process resulted in 2,445 additional training instances. Despite this augmentation strategy, the resulting dataset remained heavily imbalanced. To further mitigate this issue, we combined data augmentation with weighted loss functions, as described in the previous subsection. For reproducibility, the exact prompt used for paraphrase generation is provided in Appendix B.

6. Results

We conducted eight different experiments using two data configurations (original and augmented) and three loss functions. Due to the competition constraints, only two model configurations were

Table 2

Label distribution across annotation sets (A1, A2) and per-label F1 scores. Results are obtained with ALBERTo model with normal BCE loss and without data augmentation. Cell colors introduced as a visual aid to highlight higher (green) and lower values (red), for each column.

Code	Label	freq in A1	freq in A2	F1 (%)
AH	Ad hominem	119	115	50.86
AA	Appeal to authority	89	70	11.49
AE	Appeal to emotion	533	570	71.06
CO	Causal oversimplification	64	44	6.25
CP	Cherry picking	41	35	12.50
CR	Circular reasoning	10	6	0.00
DO	Doubt	178	188	70.32
EP	Evading the burden of proof	140	154	34.45
FA	False analogy	114	72	15.84
FD	False dilemma	30	31	0.00
FW	Flag waving	123	125	42.45
HG	Hasty generalization	176	168	33.51
LL	Loaded language	509	649	68.03
NC	Name calling or labelling	308	295	59.62
RH	Red herring	99	88	16.11
SS	Slippery slope	69	66	20.00
SL	Slogan	111	115	55.32
ST	Strawman	53	34	0.00
TC	Thought-terminating cliché	109	105	26.58
VA	Vagueness	341	475	58.23

evaluated on the official test set provided by the organizers. Overall, our experiments demonstrate the effectiveness of BERT-based models for multi-label classification in fallacy detection tasks.

Table 1 presents the results obtained under the different experimental setups. In both submissions evaluated on the official test set, our models outperform the competition’s baseline. The most notable result is achieved with the BCE loss without class weighting or data augmentation, yielding the best overall performance on the official test set: a micro-F1 score of 0.546 and a macro-F1 score of 0.326.

The second submitted model, trained using BCE loss with POS-weighting on augmented data, also outperforms the baseline across all evaluation metrics. However, its performance remains below that of the best performing configuration. These results indicate that, while class reweighting and data augmentation can provide benefits, they do not necessarily guarantee superior performance compared to training on the original data alone.

The description-aware model, trained using Focal loss on augmented data, achieved a micro-F1 score of 0.4505 on the official test set. Despite its more structured design, the results suggest that explicitly modeling label semantics does not necessarily achieve better overall performance.

Despite the clear improvements over the baseline, all submitted models remain weak in identifying fallacy labels with very low frequency, as well as fallacies whose detection depends primarily on argumentative structure rather than lexical or semantic content, such as circular reasoning and Causal oversimplification, as shown in Table 2. This indicates that rare labels and structure-driven fallacies continue to pose a significant challenge for models that rely mainly on contextual text representations.

6.1. Evaluation of different loss functions

Due to the submission constraints of the shared task, it was not possible to evaluate all loss function configurations on the official test set. Therefore, in this section, we compare different loss functions using our internally constructed test set and analyze their impact on model performance with respect to the task evaluation metrics.

We experiment with three loss formulations: BCE, weighted BCE, and Focal Loss. As reported in Table 1, all configurations substantially outperform the baseline system, regardless of the loss function

employed. Overall, the best micro-averaged F1 scores are achieved by models trained with standard BCE loss across all architectures, namely ALBERTo, UmBERTo, and XLM-RoBERTa-base, which obtain micro-F1 scores of 0.575, 0.574, and 0.567, respectively.

This behavior can be explained by the properties of the BCE loss, which assigns equal importance to all labels and therefore encourages the model to maximize the total number of correctly predicted labels, irrespective of their frequency. Given that the task’s primary evaluation metric, micro-F1, is dominated by frequent labels, this loss formulation aligns well with the evaluation objective.

In contrast, BCE with class weights and Focal Loss explicitly emphasize rare labels by increasing their contribution to the training objective. As a consequence, these loss functions generally yield higher macro-averaged F1 scores, reflecting improved performance on minority classes, but at the cost of lower micro-F1 scores. This highlights a clear trade-off between optimizing overall label-wise accuracy and improving sensitivity to infrequent fallacy types.

Given the task evaluation protocol, we therefore adopt BCE as the primary loss function in our final models.

6.2. Use of augmented data and limitations

The results reported in Table 1 indicate that augmenting the training data with synthetic paraphrases does not lead to an improvement in overall model performance. We identify two main factors that likely contribute to this outcome. First, although LLM-based paraphrasing increases the number of training instances, it can attenuate fallacy-specific rhetorical cues that are characteristic of social media discourse, such as framing strategies, slang, irony, and informal stylistic markers. As a consequence, inheriting the original labels for paraphrased instances may introduce both distribution shift and label noise. Second, fallacy detection is inherently sensitive to pragmatic and stylistic features that go beyond literal semantic equivalence. Even when a paraphrase preserves the core meaning of the original text, subtle changes in tone, emphasis, or rhetorical structure may alter the underlying fallacy signal. Together, these factors provide a plausible explanation for the lack of performance gains observed with paraphrase-based data augmentation in our experiments.

Despite these limitations, the results obtained by our best-performing model are promising, as they consistently outperform the baseline systems and other methods reported by Ramponi and Tonelli [26]. Table 2 reports the average F1 score for each individual fallacy label. As shown, the model achieves strong performance on frequent labels, such as *Appeal to emotion* (average count = 551, F1 = 71.06) and *Doubt* (average count = 183, F1 = 70.32). In contrast, performance drops substantially for less frequent fallacies, including *Red herring* (average count = 93, F1 = 16.11) and *False analogy* (average count = 93, F1 = 15.84). Moreover, the model fails to correctly identify three fallacy types, namely *Circular reasoning*, *False dilemma*, and *Strawman*.

These results highlight the intrinsic difficulty of fallacy detection, which requires models to reason beyond surface-level lexical cues. While BERT-based architectures have demonstrated strong performance across a wide range of text-classification tasks, fallacy detection often depends on higher-level discourse structure and argumentative patterns. For example, *Circular reasoning* fallacy requires understanding how claims recursively support themselves within the same text, rather than relying on isolated word-level features. As discussed by Ramponi et al. [4], successfully detecting such fallacies requires modeling global textual structure and reasoning over the full argument.

7. Conclusion

This work presents a system developed for the FadeIT task at EVALITA, which addresses multi-label fallacy classification in Italian social media posts. Our approach fine-tunes a BERT-based language model using a global multi-label classification threshold and evaluates the impact of alternative loss functions and data augmentation strategies. Experimental results indicate that models trained with paraphrase-based augmentation tend to underperform, suggesting that automatically generated paraphrases may introduce noise and distort fallacy cues present in the original data. Among the loss formulations,

Binary Cross-Entropy without class weights, trained on the non-augmented dataset, achieves the best micro-F1 score, indicating stronger overall performance on the majority labels. In contrast, BCE with class weights trained on the non-augmented dataset yields the highest macro-F1 score, suggesting that weighted loss mitigates label imbalance and benefits minority classes. This shows a clear trade-off between overall accuracy and minority-class sensitivity.

Future work may explore augmentation methods that better preserve argumentative structure, such as supporting evidence and contradiction-based variants for each instance. Another direction is to explicitly model tweet-fallacy interactions by learning representations for both the tweet and fallacy descriptions via the [CLS] embeddings and measuring their relationships using similarity-based metrics.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT-4o to correct typos and grammatical mistakes. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

References

- [1] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, 2022, pp. 7180–7198. URL: <https://aclanthology.org/2022.findings-emnlp.532/>. doi:10.18653/v1/2022.findings-emnlp.532.
- [2] M.-H. Yeh, R. Wan, T.-H. K. Huang, Cocolofa: A dataset of news comments with common logical fallacies written by llm-assisted crowds, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2024, pp. 660–677. URL: <https://aclanthology.org/2024.emnlp-main.39/>. doi:10.18653/v1/2024.emnlp-main.39.
- [3] E. Cantín Larumbe, A. Chust Vendrell, Argumentative fallacy detection in political debates, in: Proceedings of the 12th Workshop on Argument Mining, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 369–373. URL: <https://aclanthology.org/2025.argmining-1.36/>. doi:10.18653/v1/2025.argmining-1.36.
- [4] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.
- [5] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [6] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, V. Basile, Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets, in: Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019), CEUR Workshop Proceedings, 2019, pp. 312–317. URL: <https://aclanthology.org/2019.clicit-1.47/>.
- [7] J. Lanchantin, T. Wang, V. Ordonez, Y. Qi, General multi-label image classification with transformers, 2020. URL: <https://arxiv.org/abs/2011.14027>. arXiv:2011.14027.
- [8] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, H. Wang, Sgm: Sequence generation model for multi-label classification, 2018. URL: <https://arxiv.org/abs/1806.04822>. arXiv:1806.04822.
- [9] G. Da San Martino, S. Yu, A. Barrón-Cedeño, R. Petrov, P. Nakov, Fine-grained analysis of propaganda in news articles, 2019. URL: <https://arxiv.org/abs/1910.02517>. arXiv:1910.02517.

- [10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, 2018. URL: <https://arxiv.org/abs/1708.02002>. arXiv:1708.02002.
- [11] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, S. Belongie, Class-balanced loss based on effective number of samples, 2019. URL: <https://arxiv.org/abs/1901.05555>. arXiv:1901.05555.
- [12] F. Borra, C. Savelli, G. Rosso, A. Koudounas, F. Giobergia, MALTO at SemEval-2024 task 6: Leveraging synthetic data for LLM hallucination detection, in: A. K. Ojha, A. S. Doğruöz, H. Tayyar Madabushi, G. Da San Martino, S. Rosenthal, A. Rosá (Eds.), Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 1678–1684. URL: <https://aclanthology.org/2024.semeval-1.240/>. doi:10.18653/v1/2024.semeval-1.240.
- [13] H. He, Y. Bai, E. A. Garcia, S. Li, Adasyn: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 1322–1328. doi:10.1109/IJCNN.2008.4633969.
- [14] J. Li, Synthetic minority oversampling technique based on adaptive local mean vectors and improved differential evolution, IEEE Access 10 (2022) 1–1. doi:10.1109/ACCESS.2022.3187699.
- [15] M. Poliakov, N. Shvai, Missynth: Improving missci logical fallacies classification with synthetic data, 2025. URL: <https://arxiv.org/abs/2510.26345>. arXiv:2510.26345.
- [16] L. Mouchel, D. Paul, S. Cui, R. West, A. Bosselut, B. Faltings, A logical fallacy-informed framework for argument generation, 2025. URL: <https://arxiv.org/abs/2408.03618>. arXiv:2408.03618.
- [17] S. Saha, P. Yadav, L. Bauer, M. Bansal, ExplaGraphs: An explanation graph generation task for structured commonsense reasoning, in: M.-F. Moens, X. Huang, L. Specia, S. W.-t. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 7716–7740. URL: <https://aclanthology.org/2021.emnlp-main.609/>. doi:10.18653/v1/2021.emnlp-main.609.
- [18] A. Glazkova, O. Zakharova, Evaluating llm prompts for data augmentation in multi-label classification of ecological texts, in: 2024 Ivannikov Ispras Open Conference (ISPRAS), IEEE, 2024, p. 1–7. URL: <http://dx.doi.org/10.1109/ISPRAS64596.2024.10899128>. doi:10.1109/ispras64596.2024.10899128.
- [19] F. Wang, M. Beladev, O. Kleinfeld, E. Frayerman, T. Shachar, E. Fainman, K. L. Assaraf, S. Mizrahi, B. Wang, Text2topic: Multi-label text classification system for efficient topic detection in user generated content with zero-shot capabilities, 2023. URL: <https://arxiv.org/abs/2310.14817>. arXiv:2310.14817.
- [20] Ç. Çöltekin, M. Brivio, F. Can, Tübingen at politicit: Exploring SVMs, pretrained language models, and linguistic transfer for ideology detection in social media, in: M. G. Garcia, M. Passarotti (Eds.), Proceedings of the Eighth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2023), September 7–8, Parma, Italy, volume 3473 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023. URL: <https://ceur-ws.org/Vol-3473/paper10.pdf>. doi:urn:nbn:de:0074-3473-10. arXiv:3473/paper10.
- [21] H. Fallah, P. Bellot, E. Bruno, E. Murisasco, Adapting Transformers for Multi-Label Text Classification, in: CIRCLE (Joint Conference of the Information Retrieval Communities in Europe) 2022, Samatan, France, 2022. URL: <https://hal.science/hal-03727927>.
- [22] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, A. Rush, Transformers: State-of-the-art natural language processing, in: Q. Liu, D. Schlangen (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online, 2020, pp. 38–45. URL: <https://aclanthology.org/2020.emnlp-demos.6/>. doi:10.18653/v1/2020.emnlp-demos.6.
- [23] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. URL: <https://arxiv.org/abs/1810.04805>. arXiv:1810.04805.
- [24] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020.

- [25] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, 2020. URL: <https://arxiv.org/abs/1911.02116>. arXiv:1911.02116.
- [26] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [27] J. Xing, D. Luo, C. Xue, R. Xing, Comparative analysis of pooling mechanisms in llms: A sentiment analysis perspective, 2025. URL: <https://arxiv.org/abs/2411.14654>. arXiv:2411.14654.
- [28] J. L. Ba, J. R. Kiros, G. Hinton, Layer normalization, arXiv preprint arXiv:1607.06450 (2016).
- [29] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).
- [30] X. Zhang, Y. Song, F. Zuo, X. Wang, Towards imbalanced large scale multi-label classification with partially annotated labels, 2023. URL: <https://arxiv.org/abs/2308.00166>. arXiv:2308.00166.

A. Model configurations and training hyperparameters

The table below explains all model configurations that are used for experiments and official runs.

Table 3

Model configurations and training hyperparameters used in our experiments.

Component	Value
Base models	ALBERTo, UmBERTo, XLM-RoBERTa-base
Loss functions	BCEWithLogitsLoss, Weighted BCEWithLogitsLoss, Focal Loss
Optimizer	AdamW
Dropout rate	0.2
Epochs	10
Batch size	16
Learning rate	2×10^{-5}
Learning rate scheduler	Cosine schedule with linear warm-up
Warm-up steps	$0.1 \times \text{total training steps}$
Total training steps	$\lceil N_{\text{train}} / (\text{Batchsize}) \rceil \times \text{Epochs}$
Classification threshold	0.5
Focal loss α	0.25
Focal loss γ	2.0
Focal loss reduction	Mean

B. Prompt used for generating augmented data

Sei incaricato di espandere un dataset multilabel di fallacie logiche tramite **parafrasi controllate**.

OBIETTIVO

Dato un file JSON contenente un dataset annotato con fallacie logiche, devi generare nuove istanze sintetiche per aumentare la rappresentazione delle classi minoritarie.

PROCEDURA

1. Conta il numero di esempi associati a ciascuna label `labels_a1` (considera esclusivamente questo set di label).
2. Identifica le label con meno di 100 esempi.
3. Per ciascuna label rara:
 - Considera **tutti** gli esempi reali che contengono quella label.
 - Per ogni esempio reale, genera **due parafrasi differenti**.
 - Ogni parafrasi deve essere:
 - estremamente diversa in struttura e lessico;
 - breve (1-2 frasi);
 - semanticamente equivalente all'originale;
 - coerente con la fallacia associata alla label;
 - naturale e plausibile come post o commento online;
 - una riscrittura fedele, non un'estensione, interpretazione o spiegazione.
4. Aggiungi ogni parafrasi al dataset.
5. Produci il nuovo dataset JSON completo.
6. Mostra il conteggio delle label prima e dopo l'augmentation.
7. Genera un file JSON scaricabile denominato: `dataset_augmented.json`.

REGOLE DI GENERAZIONE

- *Nessun tono moralizzante.*
- *Nessun contenuto violento o sessuale superfluo.*
- *Nessun riferimento esplicito alla generazione o alla parafrasi.*
- *Mantieni il significato dell'input, modificando forma, struttura e vocabolario.*

PuDy at FadeIT: Enhancing Fine-Grained Fallacy Detection with Hierarchy-Aware Training and Contextual Enrichment

Duy Dang Phu^{1,*}, Son Bui Hong¹ and Thin Van Dang¹

¹University of Information Technology - VNUHCM, Ho Chi Minh City, Vietnam

Abstract

Fine-grained fallacy detection-identifying specific spans of argumentative errors within text is a critical yet challenging task in argument mining, characterized by data scarcity and high reasoning demands. Addressing these challenges within the **FadeIT** shared task requires not only robust extraction capabilities but also adaptation to the specific hierarchical taxonomy defined by the competition. In this paper, we adopt a span-based model architecture as the foundation for precise fallacy localization. To significantly enhance performance, we introduce two key contributions: (1) a data augmentation strategy based on *contextual enrichment* to mitigate data scarcity while preserving semantics; and (2) a *hierarchy-aware training mechanism* that utilizes label propagation to align the optimization process with the Soft F1 metric. Experimental results on the FadeIT benchmark demonstrate the efficacy of our composite approach, achieving a Soft F1 score of 50.92 and securing the Top 1 ranking in Subtask B. These findings highlight the importance of integrating data augmentation with metric-aligned training objectives in competitive argument analysis tasks.

Keywords

Span-based fallacy detection, Hierarchy-Aware Training, Data Augmentation

1. Introduction

Fallacious reasoning poses a significant challenge due to its potential to spread misinformation and distort public debates on social media. The informal, diverse, and often implicit nature of user-generated language further complicates the identification of subtle argumentative flaws. As a result, accurately detecting fallacious reasoning at a fine-grained level remains a difficult problem in computational argumentation.

Shared tasks in fine-grained semantic analysis are essential for advancing research on structured linguistic interpretation. In this context, Subtask B of the FadeIT shared task [1], organized as part of the Evalita 2026 evaluation campaign [2], focuses on span-level fallacy detection. This task is particularly challenging due to a combination of factors, including limited annotated data and a hierarchical evaluation metric.

Existing approaches often struggle to address these constraints simultaneously, particularly in terms of generalization under limited data and alignment with hierarchical evaluation metrics. To address these challenges, our approach is guided by two key design principles concerning data augmentation and training objective alignment.

First, to improve robustness and generalization under limited annotated data, we introduce a data augmentation strategy based on *Contextual Enrichment*. Unlike standard paraphrasing methods that may alter the semantics of fallacious spans, our approach strictly preserves fallacious segments while diversifying the surrounding context, encouraging the model to reduce reliance on spurious contextual cues.

Second, to better align training objectives with hierarchical evaluation metrics such as Soft F1, we incorporate a *hierarchy-aware training strategy*. By leveraging label propagation, this method enforces

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ 24520010@gm.uit.edu.vn (D. D. Phu)

🆔 0009-0006-3795-7932 (D. D. Phu); 0009-0006-7420-9212 (S. B. Hong); 0000-0001-8340-1405 (T. V. Dang)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

consistency between fine-grained fallacy labels and their superordinate categories during optimization.

The effectiveness of these design choices is demonstrated by our experimental results on the FadeIT benchmark. Our composite approach achieves strong performance and secures the **Top 1 ranking** in Subtask B, highlighting the importance of aligning data augmentation and training objectives with the semantic structure of the task and the hierarchical evaluation setting.

2. Related work

Previous work on fallacy detection has mainly focused on sentence-level classification and benchmark datasets such as Logic and LogicClimate [3] and MAFALDA [4]. More recently, fine-grained datasets with span-level annotations like FAINA have been proposed to capture nuanced fallacy expressions in social media posts [5]. Other works explore transformer-based classification in political debates [6]. Techniques to enhance reasoning, such as logical-structure-based prompts, also show performance gains [7]. However, accurately identifying fallacious spans in informal social media text remains challenging, motivating span-based models that incorporate richer contextual augmentation.

Span-based modeling. Independent of fallacy detection, span-based modeling has been widely studied across various NLP tasks. SpanBERT [8] introduces a span-aware pre-training objective by masking and predicting contiguous spans, yielding stronger representations for span selection tasks such as question answering and coreference resolution. In named entity recognition, SpanNER [9] reformulates sequence labeling as a span prediction problem, demonstrating the benefits of directly classifying candidate spans rather than assigning token-level tags. More generally, DyGIE++ [10] proposes a unified information extraction framework that enumerates, refines, and scores candidate spans for entities, relations, and events, showing that contextualized span representations can effectively capture boundary-sensitive and overlapping phenomena. These span-based paradigms provide useful inductive biases for fine-grained prediction and motivate our span-based approach to fallacy detection.

Data augmentation. Data augmentation for fine-grained sequence labeling is a non-trivial challenge, as transformations must introduce sufficient distributional diversity while strictly preserving the semantic integrity and boundary precision of annotated spans. Prior work has explored contextual augmentation through word-level substitutions based on paradigmatic relations [11], as well as LLM-based rewriting strategies for text classification tasks [12].

However, for argumentation-centric tasks such as fallacy detection, naive lexical perturbations may disrupt the underlying argument structure, leading to label noise. Recent studies highlight that models often rely on spurious correlations between labels and contextual surface cues rather than causal reasoning patterns [13], motivating augmentation strategies that explicitly disentangle invariant semantic signals from environmental noise.

3. Task Description

Subtask B of the FadeIT shared task addresses the challenge of fine-grained fallacy detection in Italian social media texts. The dataset is provided in CoNLL format with token-level annotations following a BIO tagging scheme. A key characteristic of this task is the presence of overlapping fallacies, where a single token may be associated with multiple labels corresponding to distinct fallacy types.

Formally, given an input sequence of tokens

$$\mathbf{x} = (x_1, x_2, \dots, x_n),$$

the objective is to identify all contiguous spans that exhibit fallacious reasoning and assign them the corresponding fallacy labels. Although annotations are provided at the token level, the evaluation is conducted at the span level. We therefore formulate this task as a multi-label span classification problem, which naturally supports variable-length and overlapping fallacy expressions.

To mitigate the impact of confusions among closely related fallacy categories, the shared task adopts a soft evaluation strategy based on the hierarchical taxonomy proposed by [5]. Under this scheme, predictions that correctly identify the *direct parent category* of the gold fallacy label are awarded partial credit. Consequently, our system is primarily optimized to maximize the **Soft Micro F1 score**, which explicitly accounts for hierarchical consistency between predicted and gold labels.

4. Methodology

4.1. Span-Based Modeling

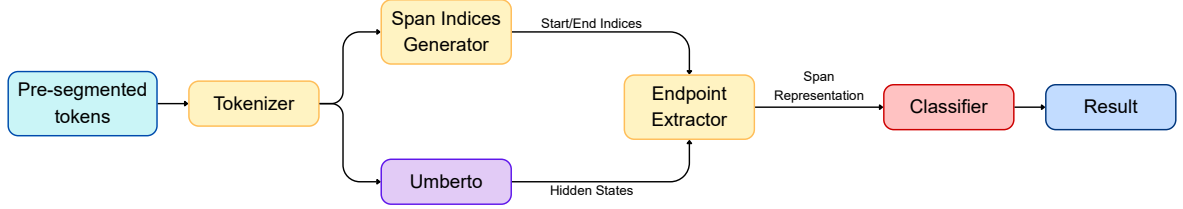


Figure 1: System Overview

We implement a span-based modeling approach leveraging UmBERTo [14] as the backbone encoder. Given the CoNLL-formatted input with pre-segmented tokens and BIO annotations, we re-tokenize the input using the UmBERTo tokenizer to obtain subword-level units, which are then fed into the encoder to produce contextualized hidden representations $\mathbf{H} = \{h_1, \dots, h_n\}$.

Following standard span-based modeling approaches [9], we enumerate all candidate spans defined by start (i) and end (j) indices. To ensure computational efficiency, we impose a maximum span length constraint such that $j - i < 20$, which covers most of gold spans in the training set while significantly reducing the computational cost.

For each valid span, we construct a span representation $s_{i,j}$ by concatenating the corresponding boundary embeddings and a learned width embedding. Specifically, the boundary embeddings h_i and h_j correspond to the contextualized token representations at positions i and j , obtained from the final hidden layer of the encoder. These embeddings capture both the internal semantics of the span and its surrounding contextual information. The learned width embedding $\mathbf{w}_{j-i+1}$ encodes the span length and provides the model with explicit structural information. The final span representation is computed as:

$$s_{i,j} = [h_i; h_j; \mathbf{w}_{j-i+1}] \quad (1)$$

where $[\cdot]$ denotes the concatenation operation and $\mathbf{w}_k$ represents the embedding for a span of width k .

The resulting span representations are passed through a linear classification layer followed by a Sigmoid activation to obtain per-label probabilities for each fallacy type.

Loss Function. Due to the severe class imbalance among fallacy types and the predominance of non-fallacious spans, we employ *Focal Loss* [15] to reduce the contribution of frequent negative examples and to focus learning on rare fallacious spans, which is particularly effective in our multi-label span classification setting.

4.2. Hierarchy-Aware Training Strategy

To better align the optimization process with the Soft F1 metric, we employ a **label propagation strategy**. Since the evaluation metric awards partial credit for superordinate categories, we modify the training targets to explicitly encode this hierarchy.

Specifically, whenever a fine-grained fallacy label is active, we enforce a soft constraint on its immediate parent category y_{parent} using the update rule:

$$y_{\text{parent}} \leftarrow \max(y_{\text{parent}}, \lambda) \quad (2)$$

where $\lambda \in [0, 1]$ is a hyperparameter acting as a minimum confidence floor. This ensures the model receives supervision for broad categories even when specific sub-types are distinguished, thereby improving robustness against hierarchical inconsistencies.

4.3. Contextual Enrichment

Motivated by the observation that span-based models may overfit to contextual lexical cues instead of argumentative structure, we propose a *Contextual Enrichment* strategy that selectively perturbs non-fallacious context while strictly preserving fallacious spans. This constraint is critical because fallacious reasoning often hinges on precise phrasing; altering the span content risks inadvertently invalidating the ground-truth label.

Formally, we decompose an input sequence x into a fallacious span s (tokens with labels B and I) and its surrounding context c (tokens with label O), such that $x = c \oplus s$. Our objective is to define an augmentation function f_{aug} that transforms only the contextual component:

$$x' = f(x) = c' \oplus s \quad (3)$$

where c' is a semantically equivalent but lexically diversified paraphrase of c , and the span s remains bit-wise immutable.

To instantiate f , we leverage the generative capabilities of the Gemini API through a controlled prompt engineering strategy. The LLM is instructed to act as a constrained re-writer, applying syntactic restructuring, synonym replacement, and mild noise injection exclusively to non-fallacious tokens, while explicitly forbidding any modification to the annotated fallacious span.

This targeted augmentation generates diverse training instances that encourage the model to focus on invariant reasoning patterns within s rather than spurious correlations in the surrounding context c , thereby improving robustness and generalization to unseen data.

4.4. Span-Relaxed Augmentation

In contrast to the strict preservation in Contextual Enrichment, this strategy permits modifying the entire sequence to introduce lexical diversity. Here, the LLM is instructed to lightly paraphrase the fallacious span s while strictly adhering to a class-consistent semantic constraint. Specifically, the transformation must preserve the exact reasoning flaw associated with the ground-truth label, thereby preventing any shift in the fallacy category.

Formally, this relaxes the token-invariance constraint, defining the transformation as:

$$x' = g(x) = c' \oplus s' \quad (4)$$

where c' denotes the rewritten context and s' represents the linguistically altered, yet semantically equivalent, version of the original fallacy.

While this approach significantly enhances the lexical diversity of the training data-preventing the model from overfitting to specific fallacy phrasings-it introduces a distinct challenge. The modification of the span boundaries necessitates distinct alignment mechanisms to ensure s' is correctly re-annotated, as the relaxation of strict constraints risks introducing label noise or subtle semantic drift.

5. Experiments

5.1. Analysis on the Development Set

In this section, all analyses are conducted on the original train-development split provided by the organizers.

5.1.1. Hierarchy-Aware Training Strategy

We investigate the impact of the proposed hierarchy-aware label propagation strategy by varying the confidence floor parameter λ . Table 1 reports the corresponding precision, recall, and Soft Micro F1 scores on the development set.

As λ increases, the model exhibits a clear trade-off between precision and recall. Specifically, higher values of λ enforce stronger supervision on parent categories, which substantially improves recall at the expense of precision. This behavior is consistent with the design of the Soft F1 evaluation metric, which assigns partial credit to correct superordinate predictions.

Notably, setting $\lambda = 1.0$ —corresponding to full activation of the parent label whenever a child label is present—yields the highest Soft Micro F1 score. This indicates that, under soft evaluation and for this dataset, aggressively propagating hierarchical supervision most effectively aligns the training objective with the evaluation criterion.

Table 1

Effect of Hierarchy-Aware Label Propagation with Different Confidence Floors (λ) on the development set.

Confidence Floor (λ)	Precision (%)	Recall (%)	Soft Micro F1 (%)
0.00	50.35	35.80	41.85
0.50	50.20 $\downarrow 0.15$	37.99 $\uparrow 2.19$	43.25 $\uparrow 1.40$
0.75	43.02 $\downarrow 7.33$	51.76 $\uparrow 15.96$	46.99 $\uparrow 5.14$
1.00	41.72 $\downarrow 8.63$	57.48 $\uparrow 21.68$	48.35 $\uparrow 6.50$

5.1.2. Contextual Enrichment vs. Span-Relaxed Augmentation

To isolate the effect of data augmentation, all experiments are conducted with the confidence floor fixed at $\lambda = 1.0$, which has been previously identified as optimal for Soft Micro F1. This controlled setting ensures that any observed performance differences can be attributed solely to the augmentation strategies. Table 2 summarizes the results under varying augmentation intensities.

Across both augmentation strategies, introducing additional augmented samples consistently improves performance over the non-augmented baseline. In particular, Recall exhibits substantial gains, suggesting that data augmentation enhances the model’s coverage of fallacious instances. These improvements indicate that moderate augmentation acts as an effective regularizer by exposing the model to a broader range of contextual realizations.

Despite these initial gains, the benefits of augmentation do not scale indefinitely. For **Contextual Enrichment**, Soft Micro F1 reaches its peak at 800 augmented samples (49.96%) before declining at 1400 samples. Notably, while Recall continues to increase, Precision decreases, leading to an overall drop in F1. This divergence suggests that excessive augmentation introduces distributional artifacts and weakens the alignment between augmented samples and the original supervision signal, thereby limiting further gains in generalization.

This effect is considerably more pronounced for **Span-Relaxed Augmentation**. Although this strategy achieves large Recall improvements—reaching 68.62% with 200 augmented samples—these gains come at the cost of a substantial Precision drop, falling below 40%. Moreover, performance degradation emerges earlier and progresses more rapidly than in Contextual Enrichment.

By directly reformulating the fallacious span itself ($s \rightarrow s'$), Span-Relaxed Augmentation increases the likelihood of semantic drift. Since fallacy identification demands high-level reasoning to discern subtle flaws, even minor linguistic perturbations in the span can unintentionally rectify the argument or shift the specific fallacy category. Consequently, the augmented instances become less faithfully aligned with their original labels.

In contrast, **Contextual Enrichment** preserves the exact wording of the fallacious span while varying only the surrounding context. This conservative design better maintains the semantic integrity of the fallacious error, resulting in more stable Precision and a higher peak Soft Micro F1 score. Given

that fallacy detection relies on fine-grained and fragile cues, this strategy proves more suitable for the task.

Table 2

Impact of Data Augmentation on Model Performance on the development set

Strategy	Samples	Precision (%)	Recall (%)	Soft Micro F1 (%)
No Augmentation	-	41.72	57.48	48.35
Contextual Enrichment	200	42.06 $\uparrow 0.34$	60.21 $\uparrow 2.73$	49.52 $\uparrow 1.17$
Contextual Enrichment	800	42.08 $\uparrow 0.36$	61.48 $\uparrow 4.00$	49.96 $\uparrow 1.61$
Contextual Enrichment	1400	40.19 $\downarrow 1.53$	61.91 $\uparrow 4.43$	48.73 $\uparrow 0.38$
Span-Relaxed Augmentation	200	38.25 $\downarrow 3.47$	68.62 $\uparrow 11.14$	49.12 $\uparrow 0.77$
Span-Relaxed Augmentation	800	39.34 $\downarrow 2.38$	63.47 $\uparrow 5.99$	48.57 $\uparrow 0.22$

5.1.3. Prompting Strategy Analysis

Given that **Contextual Enrichment** consistently outperformed **Span-Relaxed Augmentation** in previous experiments, we adopt this strategy as the primary data augmentation framework and further investigate the impact of different prompting schemes under this setting. Our initial configuration employs a **strict-instruction + few-shot** prompt, where the model is guided by a clearly defined and constrained instruction accompanied by two illustrative examples, randomly sampled from the training set, following prior findings that random few-shot selection remains a strong and reliable default for LLM-based data augmentation [16].

Building upon this baseline, we explore two alternative prompting strategies. First, we examine a **chain-of-thought (CoT)** prompting setup, in which the model is encouraged to explicitly reason through the task under a strict instruction, but without providing few-shot demonstrations. Second, we consider a **soft-instruction + few-shot** variant that remains compliant with the Contextual Enrichment constraints, but uses a less detailed and less prescriptive instruction to allow greater generative flexibility, while retaining the same two-shot examples as the baseline.

The results, summarized in Table 3, indicate that the original **strict-instruction + few-shot** configuration remains the most effective overall. Although the **soft-instruction + few-shot** strategy yields higher Recall, this improvement comes at the cost of a substantial decrease in Precision, suggesting that looser constraints introduce additional semantic noise that is detrimental to fine-grained fallacy detection. In contrast, the **CoT** prompting strategy fails to deliver consistent gains, which we attribute to the absence of few-shot exemplars that could otherwise anchor the model’s reasoning process to task-specific patterns.

Based on these findings, we retain **strict-instruction + few-shot** as the default prompting strategy for Contextual Enrichment in all subsequent experiments.

Table 3

Impact of Prompting Strategy on Model Performance on the development set.

Strategy	Precision (%)	Recall (%)	Soft Micro F1 (%)
Context Enrichment strict-instruction + few-shot	42.08	61.48	49.96
Context Enrichment strict-instruction + CoT	40.96 $\downarrow 1.12$	61.20 $\downarrow 0.28$	49.07 $\downarrow 0.89$
Context Enrichment soft-instruction + few-shot	39.80 $\downarrow 2.28$	64.94 $\uparrow 3.46$	49.35 $\downarrow 0.61$

5.2. Official Results

We submitted three system runs to the shared task. All runs employ Contextual Enrichment with a strict-instruction + few-shot prompting scheme, which consistently outperformed alternative configurations on the development set.

The first run uses 800 augmented samples, selected as a trade-off between maintaining a stable training loss and achieving strong F_1 performance on the development set. The second run adopts

the same augmentation configuration but corresponds to the checkpoint that attained the highest validation F_1 score. The third run is initialized from the first configuration and further fine-tuned for one additional epoch on the validation set.

Table 4 reports the performance of our submissions under both strict and soft evaluation settings. Among the three runs, Run 1 achieves the best Soft F_1 score (50.92), while Run 3 demonstrates a more balanced precision–recall trade-off under the strict evaluation setting.

Table 4

Performance Comparison under Strict and Soft Evaluation Modes on the official test set.

Team	Run	Strict Mode			Soft Mode		
		P	R	F_1	P	R	F_1
PuDy	1	27.68	37.83	31.97	43.76	60.88	50.92
PuDy	2	25.96	40.80	31.73	40.30	64.79	49.69
PuDy	3	29.90	32.96	31.36	45.75	52.36	48.83
TiGRO	3	47.82	37.67	42.13	51.01	40.25	44.98
TiGRO	1	38.33	40.35	39.30	42.50	45.20	43.80
TiGRO	2	38.23	40.05	39.11	42.68	44.87	43.74
RBG-AI	3	19.47	25.25	21.99	24.18	32.44	27.71
RBG-AI	2	19.21	18.24	18.71	24.67	23.96	24.31
RBG-AI	1	17.13	11.01	13.41	20.70	13.27	16.17
Baseline	–	60.94	3.05	5.80	65.97	3.28	6.25

6. Error Analysis

From Table 5, several fallacy types consistently obtain an F_1 score of 0.00 in the official results, including *Causal-oversimplification*, *Cherry-picking*, *Circular-reasoning*, *False-analogy*, *False-dilemma*, and *Strawman*. Additionally, categories such as *Slippery-slope* and *Appeal-to-authority* exhibit unstable performance, dropping to an F_1 of 0.00 in certain runs. A common pattern among these categories is the scarcity of fallacious spans in the development set whose length is less than or equal to 20 tokens. As our model follows a span-based formulation with a maximum span length of 20 tokens, these classes provide insufficient supervision signals for the model to effectively learn discriminative representations given the semantic complexity of these fallacies.

The imposed span length constraint is a deliberate design choice to balance computational efficiency and overall performance. However, fallacy types that are typically expressed through longer textual spans are consequently underrepresented during training, which leads to systematic prediction failures in the official evaluation for these categories.

Furthermore, this limitation lies beyond the scope of *Contextual Enrichment*. An essential requirement of the contextual enrichment strategy is the strict preservation of the original fallacious spans. As a result, it cannot introduce additional supervision or compensate for the scarcity of valid short-span instances without violating span boundaries.

One potential remedy would be to employ separate models or specialized components for fallacy types characterized by short versus long average span lengths. However, such a design would substantially increase training and inference costs, as well as model complexity, making it impractical under typical computational constraints. This observation highlights an inherent trade-off in span-based fallacy detection between computational feasibility and coverage of long-span fallacy types.

7. Conclusion

In this paper, we introduce a **Hierarchy-Aware Training Strategy** that aligns model optimization with soft evaluation metrics for fine-grained fallacy detection. By explicitly incorporating hierarchical label relationships into the training objective, the proposed strategy consistently improves performance

Table 5

Per-class statistics of fallacy spans on the development set (number of spans, average span length, and count of spans with length ≤ 20), together with per-class F1 scores from three independent runs on the official test set.

Fallacy Type	#Spans	Avg. Len.	#Len ≤ 20	F1 Run1	F1 Run2	F1 Run3
Ad-hominem	247	14.57	176	16.86	14.14	15.21
Appeal-to-authority	170	5.15	169	8.03	3.33	0.00
Appeal-to-emotion	1618	4.08	1598	42.47	42.89	40.85
Causal-oversimplification	113	18.51	77	0.00	0.00	0.00
Cherry-picking	76	28.18	24	0.00	0.00	0.00
Circular-reasoning	16	27.69	5	0.00	0.00	0.00
Doubt	397	15.30	304	37.04	31.72	37.07
Evading-the-burden-of-proof	318	15.47	239	12.60	9.61	5.31
False-analogy	187	20.36	107	0.00	0.00	0.00
False-dilemma	70	15.31	55	0.00	0.00	0.00
Flag-waving	316	3.27	311	33.15	35.46	42.19
Hasty-generalization	372	9.50	344	21.43	21.81	17.60
Loaded-language	2016	1.54	2010	46.90	43.48	43.55
Name-calling-or-labelling	880	1.61	880	57.42	56.28	57.06
Red-herring	199	11.51	174	5.14	4.92	4.29
Slippery-slope	140	9.48	131	31.58	0.00	0.00
Slogan	311	2.45	310	64.07	57.19	59.88
Strawman	87	36.07	17	0.00	0.00	0.00
Thought-terminating-cliches	221	4.26	220	28.86	41.24	35.16
Vagueness	1059	8.24	966	16.56	16.85	5.77

under soft evaluation, demonstrating its effectiveness in modeling graded relationships among fallacy types as defined by the label hierarchy.

In addition, we systematically investigate two data augmentation strategies tailored for fine-grained fallacy detection. Through extensive experiments, we show that while both approaches initially improve model generalization, their effectiveness differs substantially as augmentation intensity increases. In particular, **Contextual Enrichment**, which preserves the original fallacious span while varying the surrounding context, yields more stable Precision and achieves the highest overall Soft Micro F1 score. In contrast, more aggressive span-level augmentation is prone to semantic drift, leading to weaker alignment between augmented instances and their original labels.

Taken together, our findings highlight the importance of respecting label hierarchy and semantic fidelity when training models for challenging reasoning tasks such as fallacy detection. We hope this work provides practical guidance for future research on hierarchy-aware learning and data augmentation in fine-grained reasoning tasks. Notably, with these results, our system secured the 1st place in the official ranking for Subtask B of the FadeIT shared task.

8. Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT and Gemini for grammar and spelling check and paraphrase/re-wording. After using these tools, the author(s) reviewed and edited all generated content as needed and take(s) full responsibility for the publication’s content.

References

- [1] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.

- [2] F. Cutugno, A. Miaschi, A. P. Aprosio, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for Italian, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [3] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2022, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7180–7198. URL: <https://aclanthology.org/2022.findings-emnlp.532/>. doi:10.18653/v1/2022.findings-emnlp.532.
- [4] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, MAFALDA: A benchmark and comprehensive study of fallacy detection and classification, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 4810–4845. URL: <https://aclanthology.org/2024.naacl-long.270/>. doi:10.18653/v1/2024.naacl-long.270.
- [5] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.
- [6] E. Cantín Larumbe, A. Chust Vendrell, Argumentative fallacy detection in political debates, in: E. Chistova, P. Cimiano, S. Haddadan, G. Lapesa, R. Ruiz-Dolz (Eds.), Proceedings of the 12th Argument mining Workshop, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 369–373. URL: <https://aclanthology.org/2025.argmining-1.36/>. doi:10.18653/v1/2025.argmining-1.36.
- [7] Y. Lei, R. Huang, Boosting logical fallacy reasoning in LLMs via logical structure tree, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 13157–13173. URL: <https://aclanthology.org/2024.emnlp-main.730/>. doi:10.18653/v1/2024.emnlp-main.730.
- [8] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, SpanBERT: Improving pre-training by representing and predicting spans, Transactions of the Association for Computational Linguistics 8 (2020) 64–77. URL: <https://aclanthology.org/2020.tacl-1.5/>. doi:10.1162/tacl_a_00300.
- [9] J. Fu, X. Huang, P. Liu, SpanNER: Named entity re-/recognition as span prediction, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 7183–7195. URL: <https://aclanthology.org/2021.acl-long.558/>. doi:10.18653/v1/2021.acl-long.558.
- [10] D. Wadden, U. Wennberg, Y. Luan, H. Hajishirzi, Entity, relation, and event extraction with contextualized span representations, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 5784–5789. URL: <https://aclanthology.org/D19-1585/>. doi:10.18653/v1/D19-1585.
- [11] S. Kobayashi, Contextual augmentation: Data augmentation by words with paradigmatic relations, in: M. Walker, H. Ji, A. Stent (Eds.), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 452–457. URL: <https://aclanthology.org/N18-2072/>. doi:10.18653/v1/N18-2072.
- [12] H. Dai, Z. Liu, W. Liao, X. Huang, Y. Cao, Z. Wu, L. Zhao, S. Xu, F. Zeng, W. Liu, N. Liu, S. Li, D. Zhu, H. Cai, L. Sun, Q. Li, D. Shen, T. Liu, X. Li, AugGPT: Leveraging ChatGPT for Text Data Augmen-

- tation , IEEE Transactions on Big Data 11 (2025) 907–918. URL: <https://doi.ieeecomputersociety.org/10.1109/TBDATA.2025.3536934>. doi:10.1109/TBDATA.2025.3536934.
- [13] D. Kaushik, E. Hovy, Z. C. Lipton, Learning the difference that makes a difference with counterfactually augmented data, International Conference on Learning Representations (ICLR) (2020).
- [14] L. Parisi, S. Francia, P. Magnani, Umberto: an italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020.
- [15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2999–3007. doi:10.1109/ICCV.2017.324.
- [16] J. Cegin, B. Pecher, J. Simko, I. Srba, M. Bielikova, P. Brusilovsky, Use random selection for now: Investigation of few-shot selection strategies in LLM-based text augmentation, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2025, Association for Computational Linguistics, Suzhou, China, 2025, pp. 5533–5550. URL: <https://aclanthology.org/2025.findings-emnlp.296/>. doi:10.18653/v1/2025.findings-emnlp.296.

A. Prompt Templates

To ensure reproducibility, we report the full prompt templates used for data augmentation.

A.1. Contextual Enrichment (strict-instruction + few-shot)

You are an expert text augementer. Your task is to increase the contextual diversity of a given Italian sentence while keeping the meaning and the labels of non-’O’ tokens unchanged.

Instructions:

1. The sentence is provided as a list of tokens with their labels.
2. Tokens labeled ’O’ are context tokens and can be modified.
3. Tokens with other labels must remain unchanged.
4. Keep the order of tokens logical and coherent.
5. Output the new sentence as a list of tokens with the same labels.

Example Input:

<A RANDOMLY CHOSEN EXAMPLE>

Example Output:

<A RANDOMLY CHOSEN EXAMPLE>

Now apply this process to the following Italian input:

<INPUT>

A.2. Span-Relaxed Augmentation (strict-instruction + few-shot)

You are an expert in fallacies and Italian text augmentation. Your task is to rewrite a given sentence to increase diversity while preserving the specific fallacies present.

Input Format:

A list of tokens with BIO labels (e.g., B-Ad-hominem, I-Hasty-generalization, O).

Goal:

Paraphrase the sentence naturally in Italian.

Instructions:

1. Context ("O" labels):

- Significantly modify these tokens.
- You may rephrase, change sentence structure, or use synonyms to create a diverse context.

2. Fallacy Spans (B-X, I-X labels):

- You SHOULD lightly paraphrase or modify the wording of the fallacy segments to introduce variety.
- You MUST preserve the specific meaning of the fallacy type.

For example:

- * If the label is "Ad-hominem", the new phrase must still be a personal attack.
- * If the label is "Hasty-generalization", it must still be a broad claim based on insufficient evidence.

- Do NOT change the category of the fallacy.

3. BIO Tagging Integrity:

- If you rephrase a fallacy span (e.g., changing 1 word to 2 words), ensure you apply the labels correctly (B-Label for the first token, I-Label for subsequent tokens).
- Ensure the output is a valid list of JSON objects.

Example Input:

<A RANDOMLY CHOSEN EXAMPLE>

Example Output:

<A RANDOMLY CHOSEN EXAMPLE>

Now apply this process to the following Italian input:

<INPUT>

A.3. Context Enrichment (Strict Instruction + Chain-of-Thought)

You are an expert Italian Data Augmenter specialized in fallacies.

Your goal is to paraphrase the context ("O" labels) of a sentence while strictly preserving all fallacy segments.

Input Format:

A list of JSON objects with fields:

- "token"
- "label"

Your Task Process:

1. Analyze: Identify which tokens form fallacy spans (non-"O") and which form contextual parts ("O").
2. Plan: Decide how to rewrite the context tokens to change tone or style (e.g., more formal, colloquial, or emphatic), while keeping grammatical correctness in Italian.
3. Draft: Mentally construct the rewritten sentence, ensuring that all fallacy tokens remain exactly unchanged.
4. Generate: Output the final JSON list.

Constraints:

- You MUST NOT modify any token labeled with "B-" or "I-".
- You MUST return ONLY the JSON list as the final output.

Now apply this reasoning process step by step, and output the final JSON list:
<INPUT>

A.4. Context Enrichment (Soft Instruction + Few-shot)

Task: Paraphrase the contextual part of an Italian sentence without altering any fallacy triggers or labeled spans.

Rules:

- Tokens with label "O": rewrite or paraphrase.
- Tokens with labels starting with "B-" or "I-": keep exactly unchanged.

Example Input:

<A RANDOMLY CHOSEN EXAMPLE>

Example Output:

<A RANDOMLY CHOSEN EXAMPLE>

Now apply the same transformation to the following Italian input:

<INPUT>

RBG-AI at FadeIT: Prompted LLMs with Label Abstraction for Logical Fallacy Detection

Meenakshi^{1,†}, Jairam R^{1,*†}, Reshma U^{1,†}, Barathi Ganesh HB² and Michal Ptaszynski²

¹RBG AI Research, RBG.AI, SREC Incubation Center, Coimbatore, Tamil Nadu 641022, India

²Text Information Processing Lab, Kitami Institute of Technology, Kitami, Hokkaido 090-0015, Japan

Abstract

This paper describes the submission of RBG-AI to the FadeIT shared task at EVALITA 2026, addressing the identification of logical fallacies within Italian social media discussions related to migration, climate change, and public health. The task involves significant complications due to the fine-grained taxonomy of fallacies, their long-tailed label distributions, and the inter-annotator disagreement. In order to perform well under these limitations, we suggest a unified few-shot, prompt-based paradigm based on instruction-tuned large language models, with no task-specific fine-tuning. The proposed method simultaneously delivers the assistance of two complementary subtasks: (i) the identification of the type of fallacy at the sentence level by multi-label classification, and (ii) the marking of the fallacy span at the token level with BIO. We propose a label hierarchy to merge the semantically similar fallacies into larger groups. The output of this imposed space is more reliable in case of low-supervision and subjective annotation conditions. The signal design accompanied by carefully selected in-context examples is intended to be the most helpful in case of rare fallacies, multi-label instances, and annotator variation. A systematic study on various instruction-tuned models under identical settings revealed that Meta-LLaMA-3.1-8B-Instruct provided the most robust performance on both subtasks. The empirical results showed that combining label abstraction with constrained prompting greatly improved the prediction consistency, specially for low-frequency fallacies and fragmented spans. The results point out that prompt engineering and structuring of label space are crucial levers for effective fallacy detection in low-resource and high-ambiguity environments and form a practical alternative to conventional supervised fine-tuning.

Keywords

Logical fallacy detection, Prompt-based learning, Instruction-tuned language models, Annotation uncertainty, Label abstraction

1. Introduction

A fallacy is commonly defined as a type of reasoning that is persuasive but lacks logical validity [1, 2]. These forms of argumentation are relatively common in public discourse and could potentially influence opinions, when they are deliberately or mistakenly copied. For example, false dilemma arguments limit the number of alternatives, simplify complex problems, and commonly appear in political discourse [3], advertising [4], and social media platforms such as X (Twitter) [5]. They are also common in propaganda. Recently, fallacious reasoning has found a particularly significant role in polarized arguments in society, including the 2016 Brexit referendum and community discussions about COVID-19 vaccines [6, 7, 8]. This reflects the growing need for automatic fallacy detection systems in NLP to mitigate the problem of misinformation and improve critical thinking [9]. Notwithstanding the growing interest in fallacious reasoning, the problem of fallacy detection remains difficult. Most current approaches are fragmented and concentrate primarily on a specific genre or a few types of fallacies. There is no commonly accepted classification scheme [10], and fallacies can also appear jointly or overlap in the same text [11], although most datasets are simplified to a single label per segment. The subjectivity in annotation further complicates the task, as several plausible labels may exist for the same content [12].

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ meenakshi@rbg.ai (Meenakshi); jairam@rbg.ai (J. R); reshma.u@rbg.ai (R. U); hbbg.jp@gmail.com (B. G. HB); michal@mail.kitami-it.ac.jp (M. Ptaszynski)

ORCID 0009-0000-5935-9593 (J. R); 0000-0002-7172-5821 (R. U); 0000-0002-1150-2773 (B. G. HB); 0000-0002-1910-9183 (M. Ptaszynski)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The FadeIT shared task, a part of EVALITA 2026, targets these problems. It targets the fallacy detection task in Italian social media discussions on migration, climate change, and public health, using the FAINA dataset [13]. The dataset provides span-level annotations with high granularity, allows overlapping fallacies, and keeps separate annotations for different annotators [14]. In this paper, we investigate the fallacy detection performance of instruction-tuned large language models, meta-llama/Meta-Llama-3.1-8B-Instruct, in a few-shot learning scenario. By leveraging well-designed prompts and representative examples, we assess the model’s capacity to learn fallacious patterns with a high degree of annotation subjectivity. The source code for implementation and the results are publicly available at <https://github.com/rbg-research/FadeIT-EVALITA-2026>.

2. Related Works

Recent breakthroughs in large language models (LLMs) have reignited interest in applying these models for reasoning-intensive and argumentation-related tasks, such as logical fallacy detection. Previous research on models such as GPT-3 showed great success in natural language understanding and logical inference. This suggested their potential in reasoning applications [15, 16, 17, 18]. However, these models struggled to maintain coherence over longer chains of reasoning and frequently failed on multi-step inference [19]. More recently, developments with GPT-4 introduced chain-of-thought prompting to prompt step-by-step reasoning with improved outcomes for deductive reasoning performance [20]. However, recent studies have also shown that large language models continue to struggle with complex reasoning processes and are prone to error accumulation [21]. Simultaneously, the field of LLM has changed rapidly. Both closed-source and open-source models continue to deliver competitive performance [22]. Recent open-source models such as LLaMA and Qwen2.5 have shown strong results across reasoning and language understanding tasks, which narrows the gap with closed-source models [23, 24]. Other researches have also investigated the use of few-shot and prompt-based learning strategies. The researches examined the role of instruction tuning, model size, and prompt design in task adaptation without task-specific fine-tuning [25, 26]. The results of these studies stress the need for prompt engineering as an effective way to adapt LLMs to specialized reasoning tasks.

In the context of logical fallacy identification, the existing literature has primarily employed supervised transformer models. These models tend to rely on a single correct ground truth annotation, which fails to capture the subjective nature of fallacy annotation. To overcome this problem, the FAINA dataset was proposed in the research work [27]. The proposed FAINA dataset maintains annotator disagreement and supports overlapping span annotations for a total of twenty fallacy categories. In addition, it proposes an evaluation setting that supports multiple plausible gold standards and partial span alignment. The baseline experiments conducted on the FAINA dataset make it clear that multi-label and multi-task transformer models are effective reference baselines for both sentence-level and span-level fallacy detection tasks. Based on the above developments, the recent literature attempts to apply prompt-based LLMs for fallacy detection. This is especially useful in low-resource settings and ambiguous annotation settings. Few-shot prompting provides a flexible alternative to the existing fine-tuning approach, which assists the model in adapting through structured instructions and a few examples, instead of requiring a large amount of labeled data.

3. Methodology

Our goal is to address the problem of logical fallacy detection through a unified prompt-based framework that can handle both subtasks like sentence-level classification and token-level span identification. Our proposed method is motivated by three challenges associated with the problem: (i) the lack of annotated data, (ii) the subjective nature of fallacy interpretation, and (iii) the disagreement among annotators. To address these challenges, we propose a few-shot learning framework based on instruction-following large language models. The overview of our proposed method is shown in Figure 1.

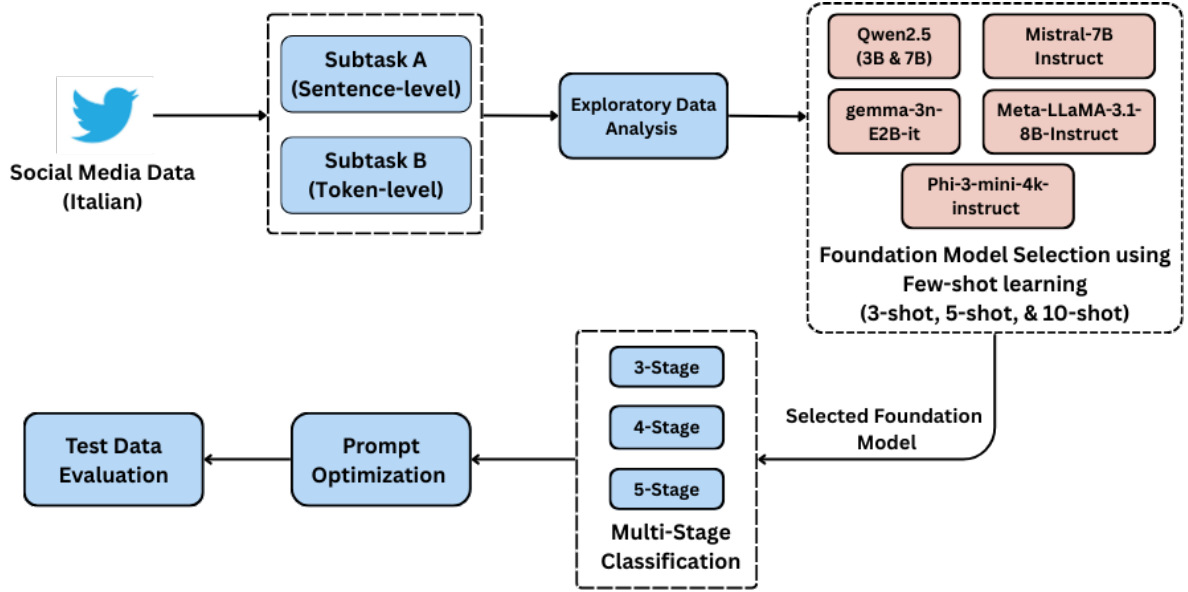


Figure 1: Overview of the proposed few-shot fallacy detection framework across both subtasks.

3.1. Task Formulation and Data Characteristics

The problem was split into two related sub-problems (subtask A and subtask B), which processed the same input texts but generated different output formats. In subtask A, fallacy detection, as a part of the FadeIT shared task [28], is formulated as a sentence-level multi-label classification problem. One sentence may demonstrate multiple types of fallacies. On the other hand, in subtask B, the fallacy detection problem is framed as a token-level sequence labeling problem. It involves detecting fallacy spans using the BIO tagging scheme. In contrast, subtask B treats fallacy detection as a token-level sequence labeling task. This requires identifying fallacy spans using the BIO tagging scheme. Although both the subtasks operate in same semantic content, they impose different modeling requirements. The sentence-level classification primarily focuses on global contextual reasoning and modeling the co-occurrence of multiple fallacy labels. In contrast, sequence labeling emphasizes accurate boundary detection and structural consistency at the token-level. Furthermore, each instance is independently annotated by two expert annotators, which naturally leads to valid disagreement in both label selection and span boundaries. The evaluation framework, following the guidelines of the official task explicitly allows for disagreements rather than forcing a single consensual annotation.

Additionally, we proceeded with an exploratory data analysis (EDA) to gain insights into the dataset characteristics. At the instance level, the files containing tab-separated values for subtask A were examined in order to check the integrity of the textual fields and annotation columns. For the purpose of clarifying the different fallacy types and topical co-occurrences, we checked the distribution of the labels assigned by the two annotators (labels_a1 and labels_a2), and the corresponding post_topic_keywords. In particular, we focused on the number of posts per topic to identify the most common subjects captured in the dataset. This task, along with data verification, also helped us identify one instance with missing value. For subtask B, we converted the token-level data released in CoNLL format into a structured dataframe for systematic analysis. This enabled us to conduct consistency checks at both token and span levels, including counting of BIO-tagged fallacy labels, analyzing span lengths, and classifying fallacy spans by topic. Overlapping or nested spans were given special attention as they are permissible under the annotation scheme and cause difficulties in sequence labeling.

The exploratory phases in combination provided a thorough view of the dataset’s most important and least important features, which then influenced the selection of model and prompt design. The analysis disclosed this is a challenging aspect for models rather than a data problem, which may influence the model performance. A more elaborate quantitative discussion of these data characteristics is presented

in Section 4.1.

3.2. Few-shot Prompt-Based Modeling Framework

In this work, task adaptation is achieved through prompt engineering and example selection, without fine-tuning the model or updating task-specific parameters. We adopt a few-shot learning framework and test 3-shot, 5-shot, and 10-shot scenarios to analyze the effects of varying lengths of prompts on the stability of predictions. The concrete prompt templates for each experimental configuration are listed in the public repository (<https://github.com/rbg-research/FadeIT-EVALITA-2026>). The exemplars are selected to achieve maximum informational diversity, including but not limited to rare categories of fallacies, multi-label instances, and differences caused by different annotators. In this prompt-based method, we design different prompt templates for each subtask according to their respective output formats. For subtask A, the prompts instruct the model to predict one or more fallacy labels from a predefined set. For subtask B, the model is asked to point out the fallacy at the token level. The predicted results are then transformed into BIO formatted tags in CoNLL format via a post-processing step that relies on span alignment and fuzzy matching. The implementation of span alignment, fuzzy matching thresholds, and logic for BIO conversion are demonstrated in the released code repository. Despite the difference in the output format of the two sub-tasks, the two templates share the same structural framework, which includes: task descriptions, definitions of labels, and output constraints. This makes the structure compact and consistent, which is helpful for the model to reason in a stable manner across the two sub-tasks without influencing the performance of the specific task.

3.3. Model Selection and Label Grouping Strategy

To investigate the potential of prompt-based few-shot learning for fallacy detection, we chose a broad range of instruction-tuned LLMs for evaluation. The choice of the models was informed by the following factors: (i) public availability and replicability, (ii) suitability for instruction tuning for few-shot prompting, and (iii) the model’s ability to support robust reasoning. The models chosen for evaluation based on the factors mentioned above include Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Mistral-7B-Instruct, Gemma-3n-E2B, Phi-3-mini, and Meta-LLaMA-3.1-8B-Instruct. The performance of all the models was evaluated under the same experimental conditions. This included the use of a common prompting template and example selection. The performance of the models was evaluated based on the following factors: (i) the coherence of multi-label predictions, (ii) the resistance to inter-annotator variability, and (iii) accuracy over spans. The best model was then selected for evaluation based on multi-stage classification.

Apart from model selection, a hierarchical grouping of labels was adopted to address the difficulties posed by the problem. The difficulties included high class imbalance and semantic overlap between the categories of fallacies. Rather than predicting on the fine-grained fallacy categories, we aggregated similar fallacies into more coarse categories based on semantic similarity and corpus-level frequency distributions (high, medium, and low frequency labels). This served to alleviate prediction ambiguity and constrain the prediction space in few-shot prompting. The label hierarchy was defined at different levels of granularity to obtain three, four, and five group settings. For each setting, the prompt explicitly constrained predictions to the predefined set of labels. This not only constrains the decision space but also enhances discrimination performance, particularly for the less numerous classes. We treated these settings as different experimental conditions and evaluated them equally for the two subtasks. This allows for a direct evaluation of the effect of label granularity on prediction stability and interpretability with the same modeling framework.

3.4. Evaluation Protocol and Prompt Refinement

As per the official task guidelines, each model produced its output separately for each annotator. The decision to use the same predicted label for both annotators is left to the participants. The participants are encouraged to model different annotators and use different predicted labels for each of them, taking

into account the variability and, at the same time, removing a source of bias introduced by the different but still valid opinions of the annotators.

The official train-dev split, provided by the organizers, included 80% of the Faina dataset for experimentation, with 20% held aside for validation. The gold-standard annotations were produced for the validation set, which enabled the direct computation of common evaluation metrics such as precision, recall, and F1-score. The validation set was used only for (i) optimizing repetition prompts, (ii) comparing model configurations, and (iii) choosing labeling strategies. The official test set was entirely held out and not accessed at any point during model development, to avoid any information leakage. The prompt candidates were subjected to a systematic evaluation that included both quantitative measures and qualitative error analysis. A key factor in prompt selection was its robustness, consistency in predictions, and adherence to the annotation guidelines during both subtasks.

4. Results and Discussions

4.1. Dataset Characteristics and Implications for Modeling

Analyzing the distributions of the labels, some critical structural issues in the FadeIT dataset can be identified. Figure 2 and 3 illustrate the distribution of the fallacy label of subtask A based on the annotation of the annotator a1 and annotator a2.

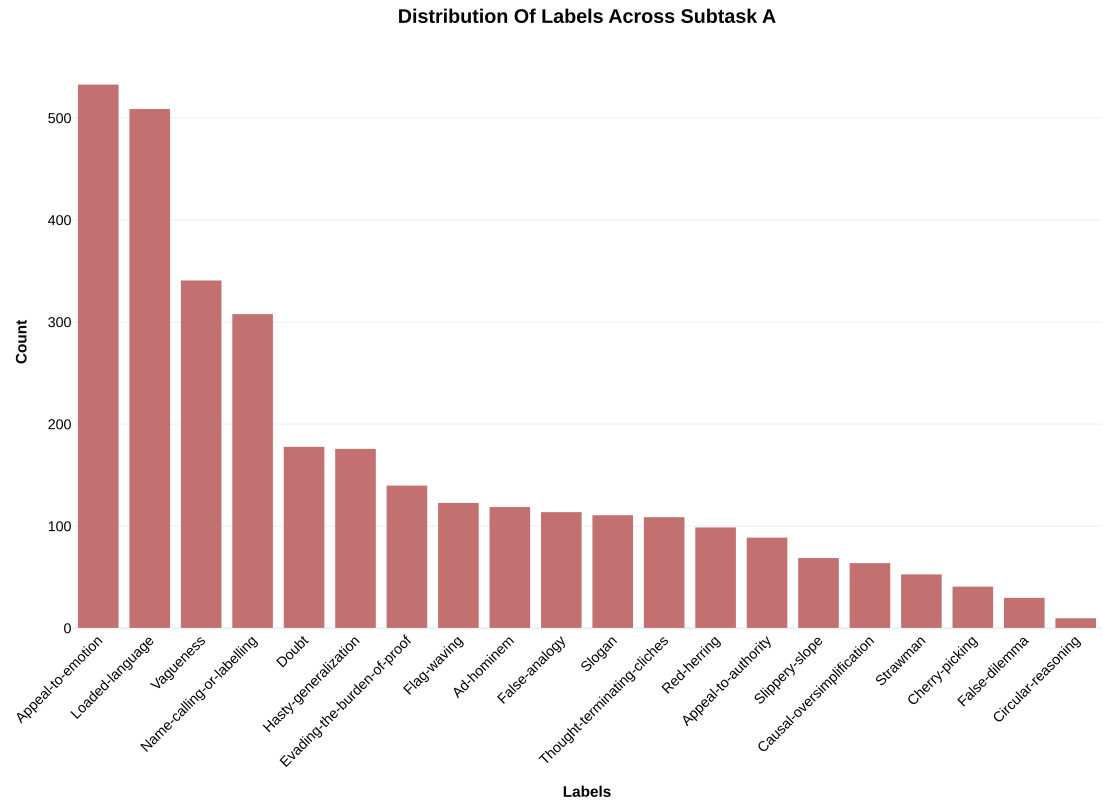


Figure 2: Label distribution for subtask A based on annotator a1, showing a strongly skewed distribution with a few dominant fallacy categories and many low-frequency classes.

The distribution is long-tailed. Some fallacies, like Appeal to Emotion, Loaded Language, and Vagueness, are more common. On the other hand, some logically subtle fallacies, such as Cherry Picking and False Dilemma, are less common. This is directly linked to the model performance in few-shot learning, where common labels are more prominent in the predictions, and less common ones are less represented in the prompt examples. The multi-label setting of subtask A makes this situation worse since multiple overlapping fallacies can be conveyed in a single sentence, making the decision

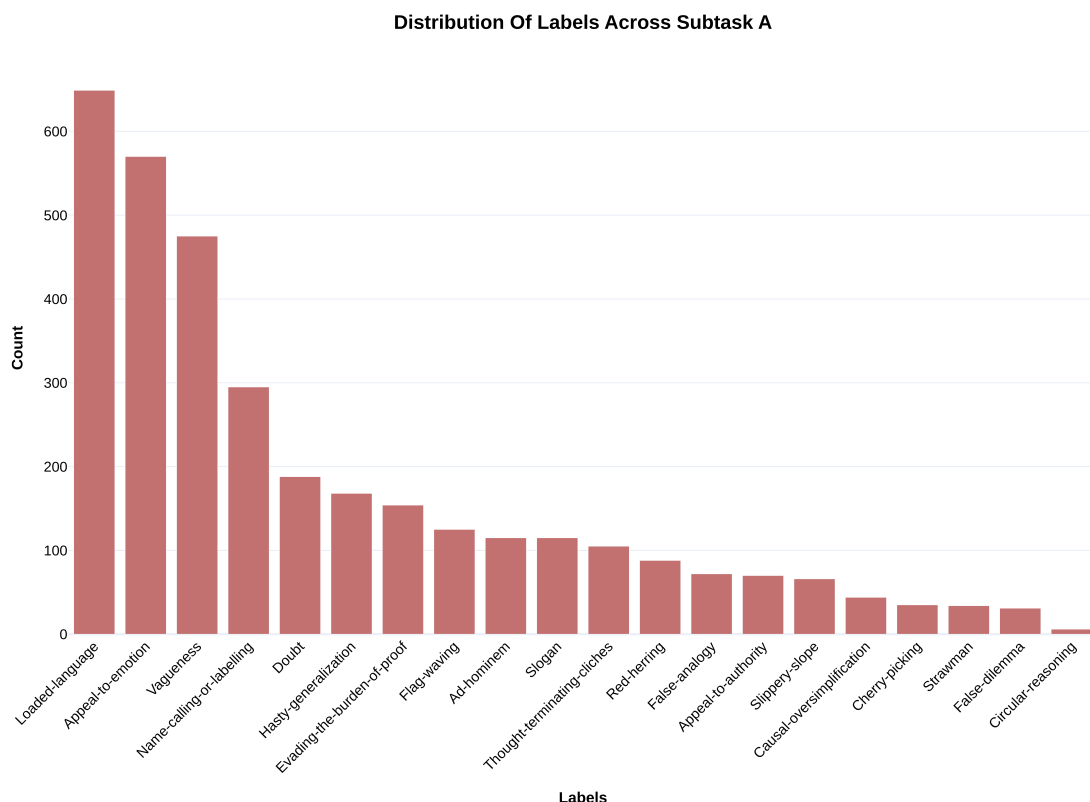


Figure 3: Label distribution for subtask A based on annotator a2, showing a strongly skewed distribution with a few dominant fallacy categories and many low-frequency classes.

boundary even more unclear. These characteristics limit the effectiveness of unconstrained multi-class classification and motivate the use of structured label-grouping strategies to stabilize the predictions.

Figure 4 and 5 shows the token-level label distribution of subtask B, excluding the overwhelmingly dominant O tag. Excluding this tag, the remaining distribution is still quite imbalanced, with most labeled tokens falling into a small minority of labels. The BIO tagging scheme exacerbates sparsity by doubling the number of fallacy categories into Band I-versions. Furthermore, span lengths vary widely across different types of fallacies. Some fallacies have short lexical cues, while others have longer argumentative spans. All these factors together contribute to the continued difficulty of token-level fallacy detection compared to sentence-level classification and the necessity of strong structural constraints in few-shot prompting.

4.2. Comparative Performance of Instruction-Tuned Models

The analysis involving the instruction-tuned language models clearly indicates a level of performance ladder achieved through the subtasks. From Table 1, the larger models, which have stronger instruction following, always outperform the smaller models, which indicates that the fallacy detection task involving subjectivity and label ambiguity benefit from larger capacity.

Meta-LLaMA-3.1-8B-Instruct scored the highest micro F1-metrics for both subtasks A and B. It demonstrated a better balance between correct positive and negative predictions (precision and recall) than any other model. The advantage of its performance is mainly evident in those situations where recall plays an important role. It is implied that the model is able to recognize fallacious reasoning which is beyond the most frequently found categories to a larger extent than the other models. On the other hand, smaller models like Qwen2.5-3B and Phi-3-mini are prone to producing unstable predictions during few-shot sessions often missing low-frequency labels or generating torn span outputs. Even though Gemma-3n-E2B-it participates actively in the competition of sentence-level classification it still

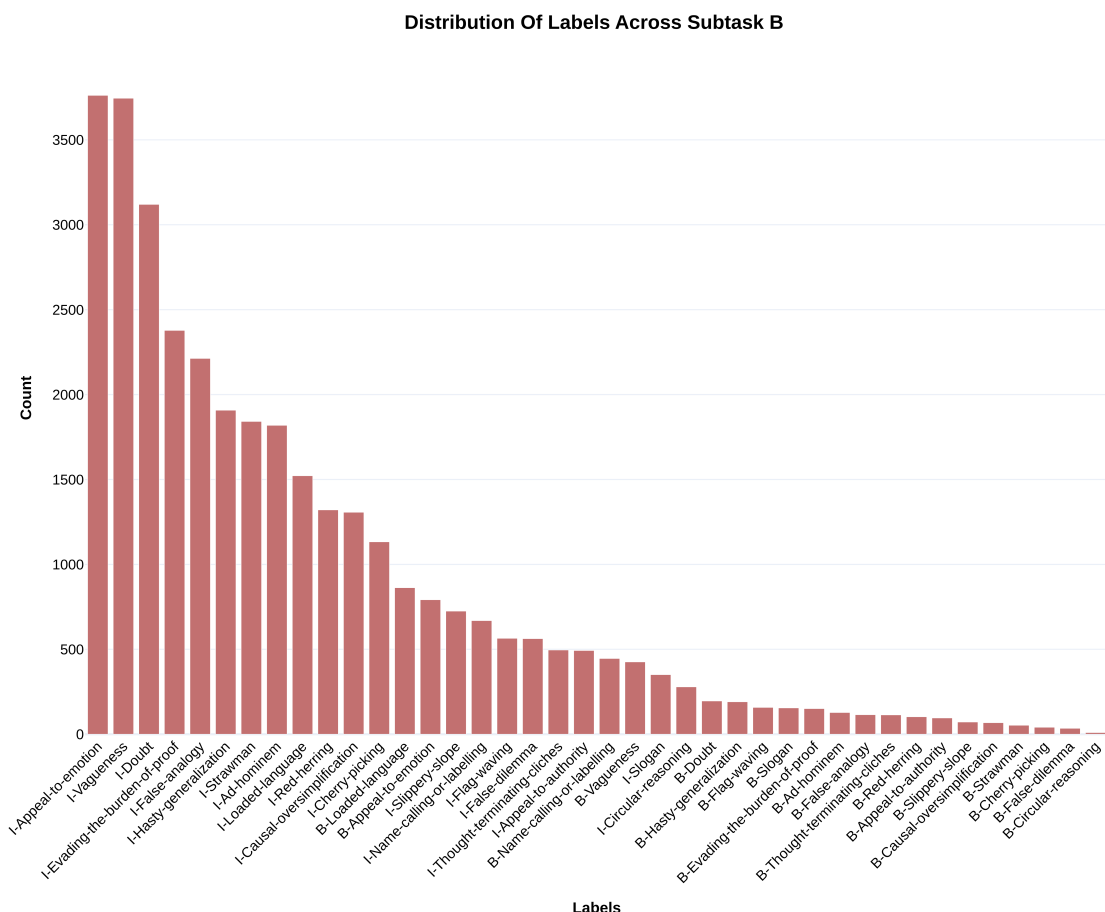


Figure 4: Distribution of BIO-tagged fallacy labels for subtask B (Annotator a1) after removing the O tag, illustrating substantial class imbalance at the token level.

Table 1

Performance comparison of instruction-tuned LLMs on Subtask A and Subtask B on the validation set (20% of train-dev split)

Model	Subtask A (micro F1)	Subtask B (micro F1, soft eval.)
Qwen2.5-3B-Instruct	13.71	2.41
Qwen2.5-7B-Instruct	17.77	7.63
Mistral-7B-Instruct-v0.3	21.39	10.11
gemma-3n-E2B-it	28.71	11.57
Meta-LLaMA-3.1-8B-Instruct	32.68	13.68
Phi-3-mini-4k-instruct	21.68	7.56

loses some points in span-level performance which indicates it is still having difficulties with fine-grained boundary detection. The findings suggest that detecting fallacies goes beyond just recognizing surface-level language features. Rather, it requires the capacity to combine annotation-specific conventions with abstract argumentative structures. As a further step, the models’ precision-recall behavior is analyzed and the trade-off between the average precision and recall of each task is visualized in Figure 6.

The model’s behavior concerning the precision-recall trade-off is very different for the two subtasks, as shown in Figure 6. In Subtask A, the models with higher capacity occupy the upper region of the trade-off area and thus sustain higher recalls with almost no compromise in precision or just proportional decrease, which explains their higher F1 scores. On the other hand, the smaller models’ behavior is scattered more, reflecting the unstable decision boundaries under the few-shot prompting condition that they are seeing. The mentioned phenomenon is even stronger in Subtask B, where

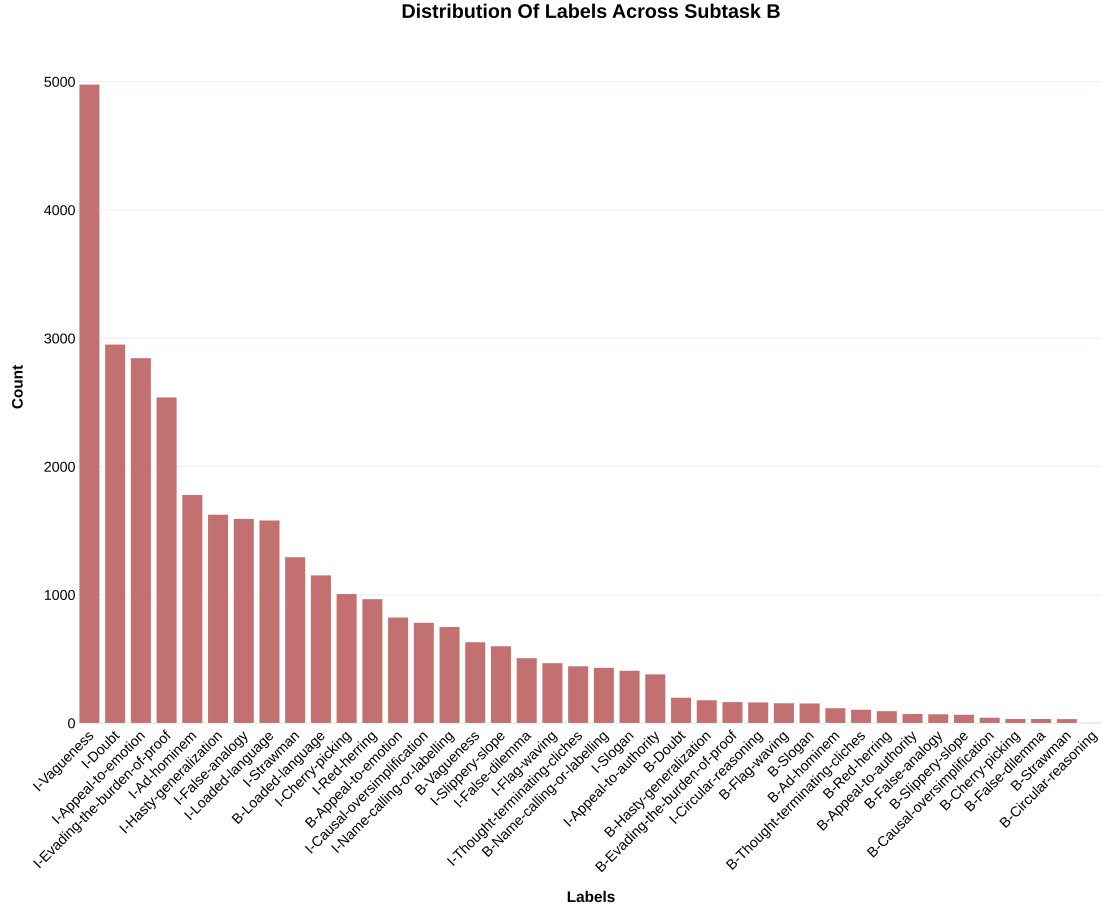


Figure 5: Distribution of BIO-tagged fallacy labels for Subtask B (Annotator a2) after removing the O tag, illustrating substantial class imbalance at the token level.

the overall recall is still low because of the sparsity and fragmentation of the span-level annotations. However, the larger instruction-tuned models give rise to better-calibrated predictions which is a sign that the effective span detection relies not only on lexical sensitivity but also on the ability to impose structural consistency within the limited output formats.

4.3. Effect of Hierarchical Label Granularity

The impact of different levels of hierarchical label abstraction on model behavior was evaluated by analyzing the subtasks through few-shot prompting, focusing mainly on stability, interpretability, and consistency of predictions in the face of annotation subjectivity and extreme class imbalance.

Considering Subtask A, dedicated to sentence-level multi-label classification, the three-stage label configuration (as described in section 3.3), where fallacies were abstracted into three groups based on corpus-level distribution and semantic relatedness attained the highest performance overall (Table 2). Grouping labels into coarser categories increased recall due to enlarging the decision space to some extent, but it very often resulted in the lowering of precision where the prediction was too generic and hence, wrong. On the contrary, the finer-grained configurations created more confusion in the multi-label situations, especially when the semantically related fallacies occurred in the same sentence. The three-stage abstraction was able to provide a balanced trade-off by limiting the output space just enough to be able to encompass overlapping fallacy types, thereby resulting in more stable and interpretable predictions. Subtask B involved identifying fallacies through BIO-based sequence labeling, and the three-stage and four-stage configurations provided similar overall performance. The four-stage configuration obtained a slight increase in soft span evaluation, mainly for short and contiguous

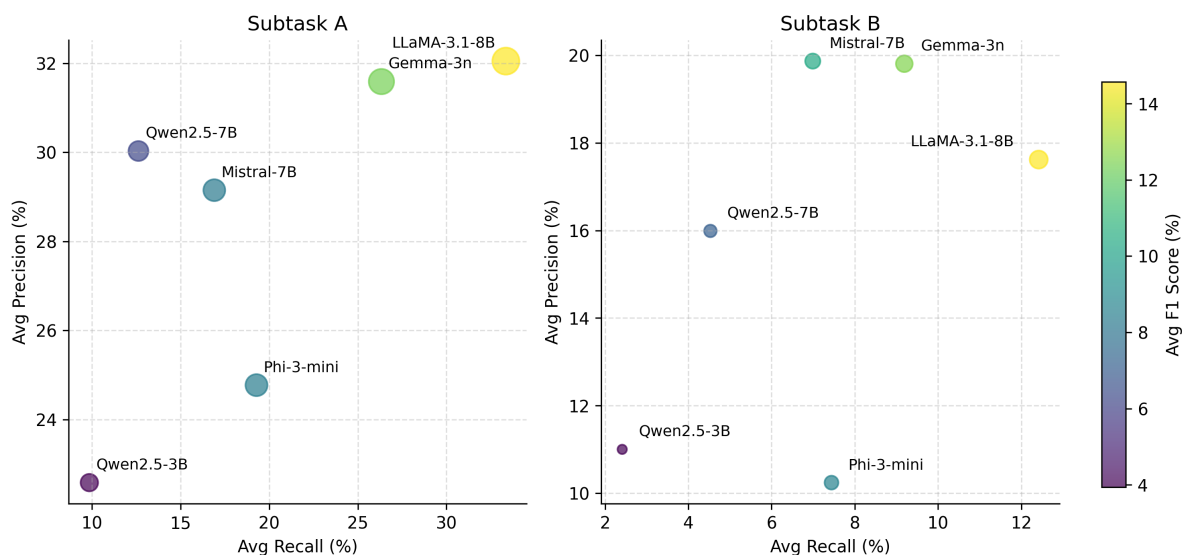


Figure 6: Precision-recall trade-off across instruction-tuned language models for Subtask A (sentence-level multi-label classification) and Subtask B (token-level span detection).

segments. Nonetheless, these advancements were not uniform across fallacy types and they were linked to a rise in fluctuation in span boundary predictions. Considering the scarcity and unevenness associated with token-level annotations, finer labels did not necessarily lead to an overall performance boost in a systematic manner.

Table 2
Effect of multistage label classification on Subtask A and Subtask B

Multi-Stage Setting	Subtask A (micro F1)	Subtask B (micro F1, soft eval.)
3-Stage	39.03	20.41
4-Stage	36.10	21.77
5-Stage	33.58	21.14

Both subtasks were considered collectively, and hence the three-stage configuration was ultimately selected as the final setting. The decision for the latter was based on the fact that although the finer abstractions provided slight advantages related to specific tasks, the overall reduction of methodological complexity and the configuration-induced biases through the use of the same label structure across the subtasks were worth it. Thus, the decision favored robustness and cross-task consistency instead of performance gains on isolated tasks. Overall, the results show that label abstraction at a moderate level is more beneficial than fine-grained partitioning in few-shot fallacy detection. Limiting the label space has a positive effect on the stability and reliability of inference, particularly in low-resource situations where there is a lot of variability in the annotations and long-tailed label distributions.

4.4. Sensitivity to Prompt Design

Prompt design is shaking out to be one of the most important factors that affect the performance of the model, particularly for subtask A. The left side of Figure 7 shows the effects of various prompt expressions on precision, recall, and micro F1-score. The prompts with too few output constraints tended to raise their recall by overpredicting the fallacy labels. The prompts that were too strict reduced the predictions of valid multi-labels, leading to a reduction in recall.

The most informative prompts were those with task instructions, definitions of label terms, and output constraints simultaneously. This approach is beneficial in ensuring that the model makes decisions within a well-defined region of the decision space, thus providing more reliable predictions

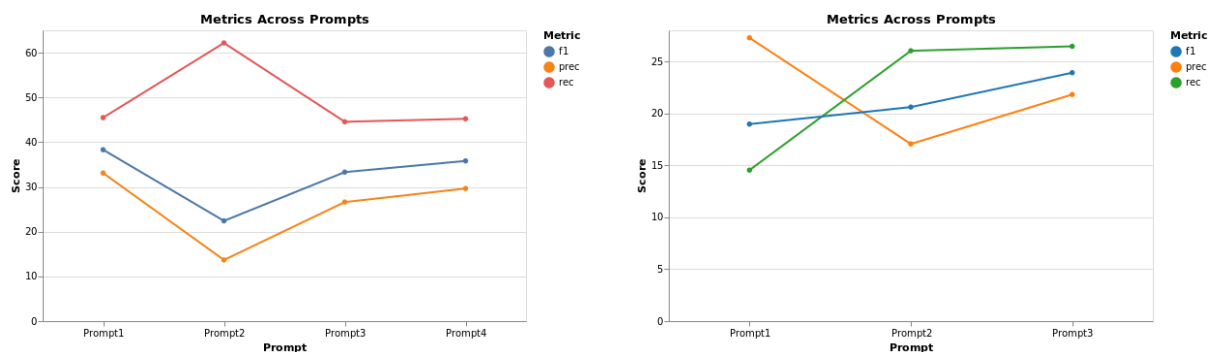


Figure 7: Impact of prompt formulation on model performance across subtasks: (left) precision, recall, and micro F1-score for sentence-level fallacy detection in subtask A; (right) micro F1-score trends for token-level fallacy detection in subtask B under soft span evaluation.

on validation sets. Of all the prompt designs considered, Prompt 1 demonstrated the smallest variability in the precision-recall curve. Hence, it is selected for further experiments.

For subtask B, the right side of Figure 7 indicates that the sensitivity to the prompt is lower in comparison; nonetheless, it is still a significant consideration. The span-level predictions are highly dependent on the accuracy of the BIO tagging task instructions and the token alignment constraints. The prompts emphasizing structural consistency and formatting rules result in more consistent spans and reduced errors in boundary fragmentation. These findings confirm the notion that prompt engineering is a core aspect of the modeling procedure rather than a secondary consideration for additional gains in few-shot fallacy detection. The full specification of all considered prompt designs, including Prompt 1 and its variations, can be found in the released code repository for complete reproducibility.

4.5. Discussion of Empirical Findings

The experimental results show that, in the FadeIT scenario, the most important factor influencing the detection of logical fallacies is the quality of the characteristics’ fit. This is more important than new design features. Moreover, problems such as disagreement between annotators, unbalanced label distributions, and overlapping semantics between fallacy categories make it difficult to apply supervised learning techniques. These factors amplify the challenge further when they are operating in low-resource scenarios. The results indicate that it is very important to keep in mind the issue of subjectivity in annotation and the scarcity of annotated data when designing models aimed at achieving robust performance.

For both subtasks, the main reason for the improvements was the implementation of techniques that helped diminishing uncertainty at the inference stage. Hierarchical label abstraction proved to be very successful in narrowing down the output space, hence limiting the influence of common fallacies and also improving the detection of those categories which are less represented in the dataset. Among the three-label configurations that were experimented with, the one which provided the most consistent compromise was the one that did not break down into too many categories. This is because making too many finely grained categories became a source of confusion during in-context learning and on the contrary, making overly coarse groups resulted in the problem of increased false positives.

The prompt engineering technique proved to be a fundamental part of the modeling process. Using prompts with clear instructions, simple label definitions, and tight output constraints led to more consistent predictions than those made with unconstrained formulations. This was especially evident in sentence-level multi-label classification. Additionally, using structurally constrained prompts was key in lowering the number of errors caused by boundary misclassifications in token-level BIO-based span detection.

The model’s capacity was another key factor that influenced its robustness. The larger instruction-tuned models, especially Meta-LLaMA-3.1-8B-Instruct, demonstrated more consistent reasoning, im-

proved generalization to low-frequency fallacies, and more coherent span predictions. Finally, the combination of multi-stage label grouping and exemplar driven prompting served as efficient strategies, which enabled stable inference under few-shot settings. Overall, these results emphasize the prompt centric, label structured paradigm as a strong alternative to task-specific fine-tuning for fallacy detection.

4.6. Official Test Set Results

Table 3

Official test set results on the FadeIT shared task

Run	Subtask A (Avg. micro F1)	Subtask B (Avg. micro F1, soft eval.)
I	38.57	16.17
II	42.07	24.31
III	40.00	27.71

Table 3 shows the official results on the FadeIT test set, which comprises 20% of the Faina dataset, as evaluated by the task organizers. All three submitted Runs uses the chosen Meta-LLaMA-3.1-8B-Instruct model under a 3-shot prompting setup, varying only in their label configuration strategy. For Run I, a flat label configuration was used, where the full fallacy labels was shown simultaneously to the model within a single prompt, allowing the model to predict directly over all labels without any hierarchical abstraction. Run II uses the proposed three-stage hierarchical label grouping strategy (Section 3.3), in which fallacies were partitioned into three subsets based on semantic abstraction and corpus-level frequency distributions. In this setting, the model was invoked three times per instance, each time restricted to one predefined subset of labels. Only fallacies from the active subset of labels were regarded as legitimate outputs, and the final label set was created by combining the predictions from the three invocation. Run III employed the same approach but with a four-stage grouping configuration, consisting of four consecutive model calls per case. As evident from Table 3, among the submitted Runs, Run II performed the best on average micro-F1 for Subtask A. Simultaneously, Run III reported the best results on Subtask B when evaluating soft spans. The above results demonstrate that although slightly more refined grouping can help with token-level span detection, moderate abstraction of labels helps with sentence-level multi-label prediction.

5. Conclusion

The current study examines the process of logical fallacy identification by means of two interrelated tasks. The research involves sentence categorization via multi-labeling and token-level span identification in a few-shot prompt-based learning setting. The findings of the research demonstrate the difficulties of logical fallacy identification. For instance, the task is challenging even without the presence of complicating factors such as class imbalance, sparsity, and overlap of annotations, as well as diverse interpretations of fallacies. In both tasks, larger language models that received training instructions performed better in terms of reasoning. They generated more coherent multi-label predictions and demonstrated higher consistency at the span level. The proposed multi-stage label grouping strategy worked well in handling long-tailed label distributions. The strategy achieved a balance between detail and stability in decision-making. Reducing uncertainty by using a smaller label set and designing appropriate prompts is critical in improving model robustness. This is especially the case in resource-scarce settings where task instructions have a significant impact on model behavior.

Future studies could investigate techniques for label abstraction. These would enable the granularity of the labels to be varied depending on the complexity of the context or the confidence level of the prediction made by the model. In addition, the use of large language models could be integrated with structured sequence labeling modules to improve performance and semantic consistency in span-based

fallacy identification. Extending the study to a multilingual and cross-domain setting would be valuable but would pose a challenge in understanding the process of generalization.

Declaration on Generative AI

During the preparation of this work, the author(s) used Generative AI in order to: Grammar and spelling check. After using these tool(s) the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] C. L. Hamblin, *Fallacies*, Advanced Reasoning Forum, 2022.
- [2] C. W. Tindale, *Fallacies and argument appraisal*, Cambridge University Press, 2007.
- [3] O. Balalau, R. Horincar, From the stage to the audience: Propaganda on reddit, in: *EACL 2021-16th Conference of the European Chapter of the Association for Computational Linguistics*, 2021.
- [4] V. Danciu, et al., Manipulative marketing: persuasion and manipulation of the consumer through advertising, *Theoretical and Applied Economics* 21 (2014) 19–34.
- [5] F. Macagno, Argumentation profiles and the manipulation of common ground. the arguments of populist leaders on twitter, *Journal of Pragmatics* 191 (2022) 67–82.
- [6] F. Zappettini, The brexit referendum: How trade and immigration in the discourses of the official campaigns have legitimised a toxic (inter) national logic, in: *"Brexit" as a Social and Political Crisis*, Routledge, 2021, pp. 23–39.
- [7] S. A. Elsayed, O. Abu-Hammad, A. B. Alolayan, Y. S. Eldeen, N. Dar-Odeh, Fallacies and facts around covid-19: the multifaceted infection, *Journal of Craniofacial Surgery* 31 (2020) e643–e644.
- [8] G. Da San Martino, S. Yu, A. Barrón-Cedeno, R. Petrov, P. Nakov, Fine-grained analysis of propaganda in news articles, in: *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 5636–5646.
- [9] U. Ecker, J. Roozenbeek, S. Van Der Linden, L. Q. Tay, J. Cook, N. Oreskes, S. Lewandowsky, Misinformation poses a bigger threat to democracy than you might think, *Nature* 630 (2024) 29–32.
- [10] H. Hansen, *Fallacies*. en zalta, 2020.
- [11] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schoelkopf, Logical fallacy detection, *arXiv preprint arXiv:2202.13758* (2022).
- [12] B. Plank, The 'problem' of human label variation: On ground truth in data, modeling and evaluation, *arXiv preprint arXiv:2211.02570* (2022).
- [13] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [14] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, *arXiv preprint arXiv:2502.13853* (2025).
- [15] J. Chen, Z. Liu, X. Huang, C. Wu, Q. Liu, G. Jiang, Y. Pu, Y. Lei, X. Chen, X. Wang, et al., When large language models meet personalization: Perspectives of challenges and opportunities, *World Wide Web* 27 (2024) 42.
- [16] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al., Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects, *Authorea preprints* 1 (2023) 1–26.
- [17] M. Zhang, J. Li, A commentary of gpt-3 in mit technology review 2021, *Fundamental Research* 1 (2021) 831–833.
- [18] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, J. Gao, Large language models: A survey, *arXiv preprint arXiv:2402.06196* (2024).

- [19] J. Yang, H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, S. Zhong, B. Yin, X. Hu, Harnessing the power of llms in practice: A survey on chatgpt and beyond, *ACM Transactions on Knowledge Discovery from Data* 18 (2024) 1–32.
- [20] T. Gou, B. Zhang, Z. Sun, J. Wang, F. Liu, Y. Wang, J. Wang, Rationality of thought improves reasoning in large language models, in: *International Conference on Knowledge Science, Engineering and Management*, Springer, 2024, pp. 343–358.
- [21] Y. Tong, D. Li, S. Wang, Y. Wang, F. Teng, J. Shang, Can llms learn from previous mistakes? investigating llms’ errors to boost for reasoning, *arXiv preprint arXiv:2403.20046* (2024).
- [22] K. He, R. Mao, Q. Lin, Y. Ruan, X. Lan, M. Feng, E. Cambria, A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics, *Information Fusion* 118 (2025) 102963.
- [23] B. Hui, J. Yang, Z. Cui, J. Yang, D. Liu, L. Zhang, T. Liu, J. Zhang, B. Yu, K. Lu, et al., Qwen2. 5-coder technical report, *arXiv preprint arXiv:2409.12186* (2024).
- [24] A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, et al., Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement, *arXiv preprint arXiv:2409.12122* (2024).
- [25] G. I. Winata, A. Madotto, Z. Lin, R. Liu, J. Yosinski, P. Fung, Language models are few-shot multilingual learners, *arXiv preprint arXiv:2109.07684* (2021).
- [26] D. Huang, X. Fu, X. Yin, H. Pen, Z. Wang, Automating maritime risk data collection and identification leveraging large language models, in: *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, 2024, pp. 433–439.
- [27] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. doi:10.18653/v1/2025.naacl-long.34.
- [28] H. Al-Omari, M. Abdullah, O. AlTiti, S. Shaikh, Justdeep at nlp4if 2019 task 1: Propaganda detection using ensemble deep learning models, in: *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, 2019, pp. 113–118.

TiGRO at FadeIT: *E Pluribus Unum* – A Multi-task Approach to Fallacy Detection and Span Identification

Stefano Atzeni^{1,2}, Gabriele Sarti², Tommaso Caselli² and Malvina Nissim²

¹University of Turin, Turin, Italy

²Rijksuniversiteit Groningen, Groningen, the Netherlands

Abstract

As a joint Turin-Groningen effort, the TiGRO team participated in the Fallacy Detection in Italian Social Media Texts (FadeIT) task at EVALITA 2026. We submitted three systems for Subtask A, which involves predicting fallacy labels across 20 classes and 2 annotators. We also submitted three systems for Subtask B, which involves predicting the text segments corresponding to the fallacies. We merged the two annotations and addressed the problem using different configurations of multi-task learning, framing Subtask A as 20 binary classification tasks and Subtask B as 20 BIO tagging tasks. Our best model jointly trains on both post-level and span-level tasks, resulting in a total of 40 tasks. Our team ranked **first for Subtask A** among fifteen submissions, achieving a micro F1 score of **56.39**, and fourth for Subtask B among nine submissions, with a micro F1 score of **44.98**. We also discuss strategies which we eventually did not adopt in our submitted runs, such as multi-lingual data augmentation and attribution-based span detection, and put forward a few ideas to enhance dataset quality.

Keywords

fallacy detection, Italian, encoder-based models, multi-task learning

1. Introduction

Argumentation and dialogue are two pillars supporting peaceful coexistence among individuals and nurturing democratic societies. However, conversational exchanges can be flawed by fallacious arguments. Fallacies are identified as instances of faulty reasoning in the construction of an argument. Superficially, they are arguments that appear to be valid while being faulty [1, 2]. Fallacies tend to be widespread across highly debated topics, whether introduced unintentionally through negligence or used intentionally to persuade audiences. Fallacies are particularly heinous in the context of political debates as they can be used as strategies to convince audiences and gain support [3], thus poisoning the public discourse and contributing to the spread of misinformation and fake news. As a matter of fact, previous work has shown that fallacies played a role in events such as the Brexit referendum [4] and in the “infodemic” crisis surrounding the COVID-19 pandemic [5].

The study of fallacies has a long tradition dating back to Aristotle and have always been contrasted to an ideal model of good arguments and reasoning [6]. The changes in human communication mediated by the growth and promotion of social media offers an excellent playground to test the functionalities of recent Natural Language Processing (NLP) systems in detecting fallacies in every day communication. The availability of a robust tool for fallacy detection in social media communication can contribute to the development of effective countering interventions against misinformation and harmful content [7, 8].

Automatic Fallacy Detection (AFD) has recently seen a surge with the availability of new datasets [9, 10, 11, 12] covering different domains such as politics and news, and LLM-based models [13, 14, 15, 16, 17, 8], including new shared task on multimodal fallacy detection [18].

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

✉ stefano.atzeni178@edu.unito.it (S. Atzeni); g.sarti@rug.nl (G. Sarti); t.caselli@rug.nl (T. Caselli); m.nissim@rug.nl (M. Nissim)

🌐 <https://gsarti.com/> (G. Sarti); <https://malvinanissim.github.io> (M. Nissim)

🆔 0000-0001-8715-2987 (G. Sarti); 0000-0003-2936-0256 (T. Caselli); 0000-0001-5289-0971 (M. Nissim)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Tasks Description The FadeIT Shared Task at EVALITA 2026 [19] is the first shared task on AFD for Italian using social media texts. The task is organized into two subtasks:

- **Subtask A: Coarse-grained fallacy detection.** Given the text of a social media post, the participants are asked to predict all the fallacies expressed in it. This is a multi-label classification task involving 20 fallacy classes.
- **Subtask B: Fine-grained fallacy detection.** Given the text of a social media post, the participants are asked to identify the text spans corresponding to each of the fallacies expressed in it. This is a multi-label sequence labeling task.

This paper presents the description of the systems our team submitted to the FadeIT shared task. Section 2 introduces the FAINA dataset used for training and evaluation. Our systems are described in Section 3, our best runs ranked first in Subtask A and fourth in Subtask B among all other participants. Results are discussed in Section 4, including a per-class analysis which highlights some limitations of the dataset. Section 5 reports on methods explored during system development but ultimately not adopted. Finally, we outline directions for future work in Section 6.

2. Dataset

The task is based on the FAINA dataset introduced by Ramponi et al. [8], which consists of 1,440 Italian Twitter/X posts collected over a four-year period (from 2019-01-01 to 2022-12-31) covering topics such as migration, climate change, and public health.

A total of 20 fallacy types were identified by harmonizing definitions and names from Da San Martino et al. [20] and Musi et al. [21], among others. The complete list of definitions and examples is provided in Ramponi et al. [8]. The dataset was annotated by two expert annotators following a prescriptive protocol [22], with multiple rounds and discussion between the annotators. As Inter-annotator Agreement (IAA) measure, the authors report γ and γ_{cat} as proposed by Mathet [23]. When compared to other IAA coefficient used in NLP, these two metrics offers insights both on how well annotations align (γ) and the extent to which annotation disagreement depends on categories rather than on the identification of spans (γ_{cat}). The reported results indicates that fallacy annotation is challenging and highly subjective with $\gamma = 0.624$ (span identification agreement) and $\gamma_{cat} = 0.544$ (span classification agreement).

Following the Perspectivist manifesto principles¹, annotation differences have not been reconciled into a single ground truth: each annotation has been considered valid, resulting in a total of 11,064 spans. Figure 1 illustrates an example of two different, but equally valid, annotations for the same post.

We were provided with 80% of the original FAINA dataset (1,152 posts) for training and validation, with the remaining 20% used as a held-out test set. We counted 4,124 unique fallacy annotations, with 3.58 fallacies per post on average. Table 1 summarizes the number and percentage of posts containing each fallacy in the released dataset. When a fallacy appears in both annotations for the same post, it is counted once. As shown in Table 1, the dataset is highly skewed, with few classes dominating the distribution.



Figure 1: Example of different annotations provided by the two annotators (en: "A flood of immigrants is coming in and nobody is talking about it. Problems keep piling up, where are we going to end up?")

¹<https://pdai.info>

Table 1

Number and percentage of posts annotated with each fallacy (posts may contain multiple fallacies).

Fallacy	Count	Percentage (%)
Loaded-language	699	60.68
Appeal-to-emotion	652	56.60
Vagueness	534	46.35
Name-calling-or-labelling	362	31.42
Hasty-generalization	225	19.53
Doubt	214	18.58
Evading-the-burden-of-proof	204	17.71
Flag-waving	158	13.72
Ad-hominem	140	12.15
Thought-terminating-cliches	138	11.98
Slogan	130	11.28
Red-herring	129	11.20
False-analogy	125	10.85
Appeal-to-authority	97	8.42
Slippery-slope	84	7.29
Causal-oversimplification	68	5.90
Strawman	58	5.03
Cherry-picking	58	5.03
False-dilemma	39	3.39
Circular-reasoning	10	0.87

3. Systems

In this section, we describe the settings and methods of the submitted systems for both subtasks. As a general strategy, we focused on identifying learning approaches that could maximize the results on both subtasks keeping in mind that, on the one hand, they are framed as separate tasks for the challenge while, on the other, they are very closely related and can inform each other. Indeed, a given text span can trigger a specific fallacy but, at the same time, knowing that a specific fallacy is present in the text can facilitate the identification of the corresponding span.

This has led us to experiment using two main modeling strategies: single-task and multi-task learning. In both cases, we decided to fine-tune models rather than opting for in-context learning with (generative) LLMs. This decision is mostly dictated by two elements: firstly, LLMs still underperform when prompted to use many classes [24], such as in this case; secondly, LLMs are known to struggle with such a complex task as previously reported by Ramponi et al. [8].

Based on the results obtained on our development split (see Training Setup), we selected six systems to submit to the official challenge, three per subtask. Table 2 summarizes the main characteristics of each system; the remainder of this section provides details and explanations of our modeling choices.

Table 2

Overview of the submitted TiGRO models.

System	Approach	Encoder	Subtask
TiGRO-A1	Single-task (one vs rest)	mmBERT	A
TiGRO-A2	Multi-task (post only)	ALBERTo	A
TiGRO-A3	Multi-task (post and span)	ALBERTo	A
TiGRO-B1	Multi-task (span only)	ALBERTo	B
TiGRO-B2	Multi-task (post and span)	ALBERTo	B
TiGRO-B3	Multi-task (post and span)	mmBERT	B

Dealing with Two Annotators Fallacy labels and spans for two different annotators are made available for the dataset, under the vision that single ground truth labels in inherently subjective tasks (such as AFD) are not desirable. Embracing this vision as well, rather than treating the two sets of annotations as separate, we took a broader view and decided to merge the annotations. In practice, if multiple labels are present, all of them are preserved, so the final label set for each post or token is the union of the labels assigned by the two annotators. At the span level, when two spans of the same fallacy class overlap but have slightly different boundaries, we keep a single span covering the union of both annotations. Besides theoretical considerations, the decision to adopt this strategy has been also dictated by the limited size of the dataset and the large number of labels where further splitting the signal across annotators substantially increases the task’s complexity. This strategy led to consistent performance improvements across all tested architectures.

Modeling Approaches One overall decision we made, independently of other implementation choices, is to address fallacy labeling (Subtask A) as a collection of binary tasks. Rather than dealing with a multi-class problem with 20 classes, we deal with 20 different binary tasks, one per fallacy. Subtask B was framed as a BIO tagging problem. As mentioned, for Task A we experimented with both single- and multi-task approaches. In the single-task setup, we fine-tuned 20 independent binary classifiers, one for each fallacy type (**TiGRO-A1**).

In multi-task learning, we experimented with leveraging interactions both across classes within Subtask A as well as between Subtask A and Subtask B. In one setting, we trained a multi-task classifier by fine-tuning 20 parallel tasks which jointly learn binary fallacy classification at the post level for Subtask A (**TiGRO-A2**). For Subtask B, we adopted the same multi-task strategy but framed the problem as BIO tagging at the span level (**TiGRO-B1**). Finally, in three of our systems, we jointly address *both* the fallacy classification at post level *and* the span detection for a total of 40 tasks (**TiGRO-A3**, **B2**, **B3**). Please note that **TiGRO-A3** and **TiGRO-B2** are exactly the same system and get a different name simply because they are used on Subtask A and Subtask B, respectively. **TiGRO-B3** differs from that model only in the pre-trained encoder used (ALBERTo vs mmBERT). While trained jointly, we do not impose any explicit dependency between the two tasks.

Encoders As base model for our systems, we experimented with several pretrained encoders. The two models used in our submissions are ALBERTo [25] and mmBERT (Multilingual Modern BERT)[26]. ALBERTo is a BERT-based model pre-trained on 191GB of Italian social media texts from the TWITA corpus [27]. mmBERT [26] is a recent multilingual encoder with 307M parameters and 22 layers, supporting more than 1,800 languages. Compared to BERT-like encoders, it introduces three main innovations: (i) a multi-stage learning paradigm composed of pre-training, mid-training, and decay phases; (ii) masked language modeling with a dynamic masking rate from 30% to 5% to enhance more nuanced language representations; (iii) dynamically varying data size and number of languages at each phase. Additionally, model temperature is dynamically adjusted to promote uniform sampling across languages and reduce bias toward high-resource ones.

In our experiments, although mmBERT is larger and more advanced, ALBERTo often performed better, likely due to its Italian specialization and Twitter pretraining, closely matching the FadeIT domain. For each submitted system we selected the encoder performing best on the development set

Training Setup We randomly split the train-dev portion released by the FadeIT shared task’s organizers into training (80%) and development (20%) sets, resulting in 921 and 231 instances, respectively, corresponding to 64% and 16% of the total FAINA dataset. We used the same fixed split to train and evaluate all setups, guiding our decisions during system development. For the final submissions, we trained our best models on all available data (1152 instances).

Multi-task training was performed using the MaChAmp v0.4 toolkit [28]. Unless otherwise specified (see Table A.1 in the Appendix), models were trained using the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$), with a batch size of 32, learning rate of 1e-4, encoder dropout of 0.2, and for 20 epochs.

Evaluation We evaluated our systems using the official evaluator provided by the organizers and optimized our runs for micro-averaged F1 score, used for the shared task ranking. Macro-averaged scores were also used for assessing and comparing systems (see Section 4). For Subtask B, the micro-averaged metrics also take into account of partial span matches, assigning credit proportionally to the overlap in terms of tokens [20]. Additionally, the evaluation distinguishes between *strict* and *soft* modes. In the soft setting, partial credit is assigned when the predicted fallacy, albeit wrong, is nevertheless related to the gold label according to the fallacy hierarchy introduced by Ramponi et al. [8] and provided by the organizers. Although aware that soft scoring is based on class relationships, we did not take this aspect into consideration when developing our systems.

4. Results and Discussion

The TiGRO family turned out to be very competitive, ranking first (#1) for Subtask A with **TiGRO-A3** and fourth (#4) for Subtask B with **TiGRO-B3**. For the latter, the official rank is based on soft scoring; when considering strict scoring, the TiGRO systems occupy the top three positions: **TiGRO-B3**, **TiGRO-B1**, **TiGRO-B2**, respectively, showing the same internal order observed for the soft scoring rank (see Table 6). Notably, our best-performing systems for both subtasks share the same architecture, namely a multi-task fine-tuned encoder jointly trained on 20 post-level and 20 span-level classification tasks. The only difference lies in the encoders used, ALBERTo and mmBERT, respectively.

4.1. Subtask A

Table 3 reports the results for Subtask A on the development and test sets. Our best-performing model on the development set, **TiGRO-A1**, did not generalize well to the test set and exhibits significant drops in both micro and macro F1, likely due to overfitting. In contrast, our weakest model on the development set, **TiGRO-A3**, achieved the highest micro-F1 on the test set, with a score of **56.39**.

Table 3

Results for Subtask A (*post-level fallacy detection*) on development and test sets. Best result per evaluation set in bold.

Model	Development				Test				Rank
	P	R	micro-F1	macro-F1	P	R	micro-F1	macro-F1	
TiGRO-A1	57.84	57.03	57.40	31.58	62.52	39.85	48.65	20.57	5
TiGRO-A2	55.19	53.78	54.46	32.03	53.43	51.48	52.41	27.94	3
TiGRO-A3	49.87	58.47	53.81	32.71	53.24	59.99	56.39	33.35	1
Baseline	32.21	11.37	16.8	3.44	38.53	14.28	20.84	5.1	

Table 4 shows **TiGRO-A3**’s per-class scores on the test set averaged across the annotators. Performance reflects the skewed class distribution in the training set: dominant classes, such as *Appeal-to-emotion*, achieve high F1 scores (75.47), while the least frequent classes, such as *Circular-Reasoning*, which appear in less than 1% of training instances, are not predicted at all.

Furthermore, the dominant classes tend to be more based on superficial cues in the text, making it easier to capture. In contrast, the minority classes mostly correspond to complex logical fallacies, which are inherently more challenging to identify. As a result, it is difficult to reliably assess how well the models actually deal with complex fallacies. As illustrated in Example 1, superficial lexical cues (highlighted in bold) characterize a highly frequent class such as *Loaded-language* while this is not the case for a minority class such as *Strawman* as shown in Example 2.

- (1) **Invasione**, sono partiti circa 900 immigrati dalla Libia: le Ong li stanno portando qui
(en: **Invasion**: 900 immigrants depart from Libya; NGOs are bringing them here)

- (2) [USER] propone di distribuire i vaccini “anche in base al Pil regionale”. In pratica, per lei le regioni più ricche dovrebbero essere più protette dal virus rispetto a quelle più povere. Quale sarà la prossima idea? Vaccinare i bianchi e non i neri? Vergogna
(en: [USER] proposes distributing vaccines “also based on regional GDP”. In short, for her, richer regions should be better protected from the virus than poorer ones. What will the next idea be? Vaccinate white people but not black people? Shame.)

Table 4

TiGRO-A3 per-class performance on test set, averaged across annotators, ordered by F1 score, with the percentage of training posts containing each class.

Fallacy	Precision	Recall	F1	Train (%)
Appeal-to-emotion	64.04	91.86	75.47	56.60
Loaded-language	58.33	92.16	71.26	60.68
Doubt	65.38	69.19	67.19	18.58
Name-calling-or-labelling	61.17	68.95	64.73	31.42
Slogan	72.73	57.14	64.0	11.28
Vagueness	50.0	73.94	58.9	46.35
Ad-hominem	56.9	48.53	52.37	12.15
Flag-waving	41.3	32.24	36.21	13.72
Hasty-generalization	34.38	37.71	35.91	19.53
Evading-the-burden-of-proof	29.76	33.55	31.53	17.71
Thought-terminating-cliches	28.57	26.7	27.59	11.98
Slippery-slope	37.5	18.75	25.0	7.29
False-analogy	26.92	13.95	18.18	10.85
Red-herring	17.39	14.67	15.88	11.20
Cherry-picking	25.0	7.14	11.11	5.03
Appeal-to-authority	10.0	4.95	6.6	8.42
Causal-oversimplification	10.0	3.33	5.0	5.90
Strawman	0.0	0.0	0.0	5.03
False-dilemma	0.0	0.0	0.0	3.39
Circular-reasoning	0.0	0.0	0.0	0.87

This phenomenon raises concerns about the reliability of micro F1 as an evaluation metric in highly skewed datasets. Although **TiGRO-A3** achieves a micro F1 of 56.39 on the test set, it performs poorly on most classes, highlighting how the overall score is dominated by the easiest, majority classes. To further validate this observation and assess whether the high performance on frequent classes might be due mostly to exploitable surface patterns, we trained a linear Support Vector Machine (SVM) classifier as a baseline. We used the `scikit-learn` implementation with default parameters [29] and `xlm-r-distilroberta-base-paraphrase-v1` sentence embeddings [30]. Since the test set gold labels were not released, we report results on the development set to ensure a fair comparison.

The SVM achieves a micro F1 of 47.38, notably close to **TiGRO-A3**’s 53.81 on the development set (see Table 3). However, macro F1 provides a fairer assessment of model capabilities with **TiGRO-A3** achieving 32.71 compared to only 15.04 for the SVM. More revealing insights emerge from the per-class analysis. On the development set, **TiGRO-A3** achieves a per-class F1 score of at least 50% for only 5 out of 20 fallacy classes. Notably, these are the same classes for which the SVM obtains a non-zero F1 score. Table 5 reports the development set results for these top five classes for both SVM and **TiGRO-A3**. Performance on the majority classes is similar for both models, confirming that these classes are inherently easier to detect even without complex language understanding, likely due to exploitable lexical biases in the dataset. In contrast, the SVM fails entirely on all other fallacies, whereas **TiGRO-A3** captures several mid-difficulty classes, although with limited performance. This indicates that the more complex model does identify some patterns beyond surface cues, making it difficult to reliably assess how well it actually “understands” complex fallacies.

Table 5

Performance of SVM and **TiGRO-A3** on the top five fallacy classes on development set. All remaining classes not shown achieve a F1 score of 0 for SVM and below 50% for TiGRO-A3.

Fallacy	SVM			TiGRO-A3		
	P	R	F1	P	R	F1
Doubt	89.58	54.56	67.79	73.75	74.84	74.26
Loaded-language	54.24	82.59	65.30	55.90	91.71	69.28
Appeal-to-emotion	58.03	72.59	64.47	57.52	80.43	67.04
Name-calling-or-labeling	52.56	39.87	45.33	51.35	73.85	60.57
Vagueness	50.46	68.95	57.89	48.39	75.37	58.56

Performance per Annotator During our experiments on development set we observed that the results for the second annotator are consistently higher across all setups. This is also observed on the test set, where **TiGRO-A3** achieves an F1 of 58.68% on Annotator2 and of 54.10% on Annotator1. This difference may be due to the first annotator providing noisier signals. Combining the annotations likely helped stabilize the training process and improve overall performance.

4.2. Subtask B

Table 6 present the results for Subtask B on the development and test sets. Our best model on the development set, **TiGRO-B3**, also obtained the highest performance on the test set among our submissions, with a Soft-F1 score of **44.98**. In addition, it achieved the highest Strict-F1 score among all submitted systems, reaching 42.13.

Table 6

Results for Subtask B (*span-level fallacy detection*) on development and test sets, using micro averaged scores. Best result per evaluation set in bold.

Model	Development						Test							
	Strict			Soft			Strict				Soft			
	P	R	micro-F1	P	R	micro-F1	P	R	micro-F1	Rank	P	R	micro-F1	Rank
TiGRO-B1	37.61	32.37	34.79	41.87	36.80	39.17	38.33	40.35	39.30	2	42.50	45.20	43.80	5
TiGRO-B2	37.59	33.14	35.20	41.50	37.20	39.24	38.23	40.05	39.11	3	42.68	44.87	43.74	6
TiGRO-B3	44.45	31.95	37.18	47.81	35.13	40.50	47.82	37.67	42.13	1	51.01	40.25	44.98	4
Baseline	90.49	0.77	1.53	90.49	0.77	1.53	60.94	3.05	5.80		65.97	3.28	6.25	

Table 7 shows TiGRO-B3 per-class scores on the test set, averaged across the annotators, together with span frequency and average span length in our unified version of train-dev set. In this case, span frequency appears to be less predictive of model performance, whereas span length seems to play a more important role. For instance, *Slogan* achieves strong results (F1 71.28) while representing only 2.73% of the training spans, but it has a relatively short average span length of 4.32 tokens, which may facilitate more accurate predictions.

Classes with the strongest performance in Subtask A tend to be associated with shorter spans, whereas the most difficult classes often involve much longer spans, sometimes even covering the entire post. A notable exception is *Doubt*, which performs well (F1 64.80) despite having a long average span length of 17.16 tokens and representing only 4.02% of all spans. In contrast, *Appeal-to-Authority* performs poorly (F1 12.90) despite a relatively short average span length of 6.19 tokens.

Table 7

TiGRO-B3 per-class performance on test set, averaged across annotators, ordered by F1 score, with the percentage of training spans and average span length.

Fallacy	Precision	Recall	F1	Train (%)	Avg span length
Slogan	71.67	70.93	71.28	2.73	4.32
Doubt	56.95	75.21	64.80	4.02	17.16
Appeal-to-emotion	53.94	53.93	53.90	17.85	5.74
Name-calling-or-labelling	68.51	43.67	53.27	9.84	2.71
Loaded-language	47.37	46.66	46.90	23.12	2.77
Thought-terminating-cliches	34.12	37.18	35.55	2.52	5.44
Slippery-slope	80.00	21.59	34.00	1.54	10.70
Vagueness	34.57	32.03	32.66	13.03	9.56
Flag-waving	47.22	21.62	29.66	3.46	4.66
Ad-hominem	44.87	18.25	25.91	2.62	17.07
Evading-the-burden-of-proof	28.03	10.25	15.01	3.97	16.52
Appeal-to-authority	41.67	7.69	12.90	1.90	6.19
Cherry-picking	48.48	7.14	12.45	1.03	29.21
False-analogy	26.79	7.19	11.27	2.23	20.93
Hasty-generalization	18.52	7.85	11.03	4.38	11.01
Red-herring	11.11	3.49	5.30	2.48	12.79
False-dilemma	0.00	0.00	0.00	0.80	16.96
Causal-oversimplification	0.00	0.00	0.00	1.28	20.78
Strawman	0.00	0.00	0.00	1.03	37.79
Circular-reasoning	0.00	0.00	0.00	0.18	28.90

5. Methods Not Adopted

During system development, we experimented with several alternative approaches to improve performance. While not all succeeded, discussing these attempts may benefit future work.

5.1. Data Augmentation

To address the issue of data underrepresentation, we attempted to expand the training dataset by integrating external resources. This is a well-known challenge in fallacy detection, as there is no standard taxonomy, definitions, or annotation guidelines for fallacies.

We focused on the MAFALDA dataset [31], since its fallacy classes are partially aligned with those in our task, making label mapping feasible. The dataset consists of 200 annotated English examples from different sources, including Reddit discussions, toy examples, and political debates.

We first trained the model on a multi-task setup using mmBERT (see TiGRO A2), leveraging its multilingual capabilities; however, this did not lead to noticeable performance improvements. As a second attempt, we also automatically translated the MAFALDA examples into Italian using Claude Sonnet 4.5, so that we could use the ALBERTo encoder which overall was offering better performances on our development set. This approach however proved unsuccessful to the point that we observed a decrease in performance. This degradation is likely due to (i) domain mismatch as the additional data differ substantially in style from Twitter posts, and (ii) differences in the fallacy class definitions leading to different annotation guidelines.

5.2. Span Extraction with Attribution

For addressing Subtask B, our initial approach aimed to build a performative post-level classifier for the 20 fallacy types (Subtask A), followed by span extraction using attribution methods. In other words, assuming that the classifier correctly predicts the post-level class, we can analyze which tokens the model focuses on to make that prediction and extract the most relevant ones.

Due to weak performance on Subtask A across several classes, this approach was not feasible. However, we still experimented with the highest-performing class *Appeal to Emotion* as a proof-of-concept for this idea. We used the LXT library, which provides an efficient attribution method based on AttnLRP [32], a backpropagation-based technique that corrects gradient flow through non-linearities. After computing token-level relevance scores, we selected tokens whose relevance exceeded a threshold defined as θ standard deviations above the mean. The parameter θ was tuned to maximize macro F1 on a BIO token-level evaluation for the considered class.

Overall, although the attribution method often highlights important words and can be useful as an explainability tool for human inspection, our preliminary experiments on automatic span extraction were inconclusive and yielded poor results. Some qualitative examples are reported on Figure B.1 in Appendix B. Further investigation is needed to include more classes and design sophisticated span extraction strategies that handle noisy relevance and adapt to gold span annotations.

6. Conclusions and Future Work

This paper presented the TiGRO family systems for the FadeIT task at EVALITA 2026. Our main focus was Subtask A, where TiGRO-A3 achieved first place (micro F1 of 56.39), using a multi-task approach that framed Subtask A as 20 binary classification tasks and Subtask B as 20 BIO tagging tasks, jointly training a total of 40 classifiers.

For Subtask B, our initial idea to use attribution methods to extract spans once obtained the fallacy classes from the predictions in Subtask A resulted unfeasible. This was due to poor performance on most classes in Subtask A. Instead, our submitted best model used the same multi-task approach as Subtask A, ranking fourth in soft evaluation (micro F1 44.98) and first in strict evaluation (micro F1 42.13).

Performance analysis per class revealed substantial differences across classes, reflecting the highly skewed distribution of the dataset. In addition to frequency in the training set, some classes appeared to be inherently easier, detectable even by a simple SVM, while others performed poorly or were not predicted at all, particularly the more complex logical fallacies (e.g., *Strawman* or *Circular-reasoning*), which are less likely to rely on surface clues only.

Future work should focus on extending the dataset, possibly incorporating examples from other domains to reduce biases inherent to the style of social media messages, which favors brief, provocative messages over complex arguments. Annotation could also be improved: we found that fitting to individual annotators' styles introduced noisy signals, whereas treating each annotation as equally valid significantly improved predictions. More structured guidelines with formal definitions of fallacies, following Helwe et al. [31], could improve consistency and allow the inclusion of additional annotators.

Finally, although attribution-based span extraction was not feasible in this setting, we plan to further develop this method for tasks where text-level labels are easier to obtain than precise span annotations.

Acknowledgments

We acknowledge the Erasmus+ program for funding SA's research visit at the University of Groningen, making this collaboration possible. We thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high performance computing cluster. This work made also use of the Dutch national e-infrastructure with the support of NWO Small Compute applications grant no. EINF-12946. We also thank the NWO Sectorplan programme (Humane AI) which partially supports MN.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly in order to: Grammar and spelling check, Text Translation, Improve writing style. After using these services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] D. Walton, *Fundamentals of Critical Argumentation*, Cambridge University Press, 2006.
- [2] C. L. Hamblin, *Fallacies*, Advanced Reasoning Forum, Socorro, USA., 2022.
- [3] P. Goffredo, S. Haddadan, V. Vorakitphan, E. Cabrio, S. Villata, Fallacious argument classification in political debates, in: L. D. Raedt (Ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, International Joint Conferences on Artificial Intelligence Organization*, 2022, pp. 4143–4149. URL: <https://doi.org/10.24963/ijcai.2022/575>. doi:10.24963/ijcai.2022/575, main Track.
- [4] F. Zappettini, The brexit referendum: how trade and immigration in the discourses of the official campaigns have legitimised a toxic (inter)national logic, *Critical Discourse Studies* 16 (2019) 403–419. URL: <https://doi.org/10.1080/17405904.2019.1593206>. doi:10.1080/17405904.2019.1593206. arXiv:<https://doi.org/10.1080/17405904.2019.1593206>.
- [5] S. A. Elsayed, O. Abu-Hammad, A. B. Alolayan, Y. S. Eldeen, N. Dar-Odeh, Fallacies and facts around covid-19: The multifaceted infection, *Journal of Craniofacial Surgery* 31 (2020). doi:10.1097/scs.0000000000006752.
- [6] H. Hansen, Fallacies, in: E. N. Zalta, U. Nodelman (Eds.), *The Stanford Encyclopedia of Philosophy*, Fall 2024 ed., Metaphysics Research Lab, Stanford University, 2024.
- [7] U. Ecker, J. Roozenbeek, S. Van Der Linden, L. Q. Tay, J. Cook, N. Oreskes, S. Lewandowsky, Misinformation poses a bigger threat to democracy than you might think, *Nature* 630 (2024) 29–32.
- [8] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.
- [9] P. Goffredo, S. Haddadan, V. Vorakitphan, E. Cabrio, S. Villata, Fallacious argument classification in political debates, in: *Thirty-First International Joint Conference on Artificial Intelligence {IJCAI-22}*, International Joint Conferences on Artificial Intelligence Organization, 2022, pp. 4143–4149.
- [10] P. Goffredo, M. Chaves, S. Villata, E. Cabrio, Argument-based detection and classification of fallacies in political debates, in: H. Bouamor, J. Pino, K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Singapore, 2023, pp. 11101–11112. URL: <https://aclanthology.org/2023.emnlp-main.684/>. doi:10.18653/v1/2023.emnlp-main.684.
- [11] Z. Jin, A. Lalwani, T. Vaidhya, X. Shen, Y. Ding, Z. Lyu, M. Sachan, R. Mihalcea, B. Schölkopf, Logical fallacy detection, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2022*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 7180–7198. URL: <https://aclanthology.org/2022.findings-emnlp.532/>. doi:10.18653/v1/2022.findings-emnlp.532.
- [12] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, MAFALDA: A benchmark and comprehensive study of fallacy detection and classification, in: K. Duh, H. Gomez, S. Bethard (Eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 4810–4845. URL: <https://aclanthology.org/2024.naacl-long.270/>. doi:10.18653/v1/2024.naacl-long.270.
- [13] T. Alhindi, T. Chakrabarty, E. Musi, S. Muresan, Multitask instruction-based prompting for fallacy recognition, in: Y. Goldberg, Z. Kozareva, Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022, pp. 8172–8187. URL: <https://aclanthology.org/2022.emnlp-main.560/>. doi:10.18653/v1/2022.emnlp-main.560.
- [14] F. Pan, X. Wu, Z. Li, A. T. Luu, Are LLMs good zero-shot fallacy classifiers?, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural*

- Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 14338–14364. URL: <https://aclanthology.org/2024.emnlp-main.794/>. doi:10.18653/v1/2024.emnlp-main.794.
- [15] Y. Li, D. Wang, J. Liang, G. Jiang, Q. He, Y. Xiao, D. Yang, Reason from fallacy: Enhancing large language models’ logical reasoning through logical fallacy understanding, in: K. Duh, H. Gomez, S. Bethard (Eds.), Findings of the Association for Computational Linguistics: NAACL 2024, Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 3053–3066. URL: <https://aclanthology.org/2024.findings-naacl.192/>. doi:10.18653/v1/2024.findings-naacl.192.
 - [16] E. Cantín Larumbe, A. Chust Vendrell, Argumentative fallacy detection in political debates, in: E. Chistova, P. Cimiano, S. Haddadan, G. Lapesa, R. Ruiz-Dolz (Eds.), Proceedings of the 12th Argument mining Workshop, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 369–373. URL: <https://aclanthology.org/2025.argmining-1.36/>. doi:10.18653/v1/2025.argmining-1.36.
 - [17] J. Jeong, H. Jang, H. Park, Large language models are better logical fallacy reasoners with counterargument, explanation, and goal-aware prompt formulation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Findings of the Association for Computational Linguistics: NAACL 2025, Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 6918–6937. URL: <https://aclanthology.org/2025.findings-naacl.384/>. doi:10.18653/v1/2025.findings-naacl.384.
 - [18] E. Mancini, F. Ruggeri, S. Villata, P. Torroni, Overview of mm-argfallacy2025 on multimodal argumentative fallacy detection and classification in political debates, in: Proceedings of the 12th Argument mining Workshop, 2025, pp. 358–368.
 - [19] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
 - [20] G. Da San Martino, A. Barrón-Cedeño, P. Nakov, Findings of the NLP4IF-2019 shared task on fine-grained propaganda detection, in: A. Feldman, G. Da San Martino, A. Barrón-Cedeño, C. Brew, C. Leberknight, P. Nakov (Eds.), Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, Association for Computational Linguistics, Hong Kong, China, 2019, pp. 162–170. URL: <https://aclanthology.org/D19-5024/>. doi:10.18653/v1/D19-5024.
 - [21] E. Musi, M. Aloumpi, E. Carmi, S. Yates, K. O’Halloran, Developing fake news immunity: fallacies as misinformation triggers during the pandemic, *Online Journal of Communication and Media Technologies* 12 (2022).
 - [22] P. Röttger, B. Vidgen, D. Hovy, J. Pierrehumbert, Two contrasting data annotation paradigms for subjective NLP tasks, in: M. Carpuat, M.-C. de Marneffe, I. V. Meza Ruiz (Eds.), Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Seattle, United States, 2022, pp. 175–190. URL: <https://aclanthology.org/2022.naacl-main.13/>. doi:10.18653/v1/2022.naacl-main.13.
 - [23] Y. Mathet, The agreement measure γ_{cat} a complement to γ focused on categorization of a continuum, *Computational Linguistics* 43 (2017) 661–681. URL: https://doi.org/10.1162/COLI_a_00296. doi:10.1162/COLI_a_00296. arXiv:https://direct.mit.edu/coli/article-pdf/43/3/661/1808405/coli_a00296.pdf.
 - [24] A. Muti, C. Emmery, D. Nozza, A. Barrón-Cedeño, T. Caselli, The “r” in “woman” stands for rights. auditing LLMs in uncovering social dynamics in implicit misogyny, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2025, Association for Computational Linguistics, Suzhou, China, 2025, pp. 5462–5479. URL: <https://aclanthology.org/2025.findings-emnlp.292/>. doi:10.18653/v1/2025.findings-emnlp.292.
 - [25] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, V. Basile, ALBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets, in: Proceedings of the

- Sixth Italian Conference on Computational Linguistics (CLiC-it 2019), volume 2481, CEUR, 2019. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85074851349&partnerID=40&md5=7abed946e06f76b3825ae5e294ffac14>.
- [26] M. Marone, O. Weller, W. Fleshman, E. Yang, D. Lawrie, B. V. Durme, mmbert: A modern multilingual encoder with annealed language learning, 2025. URL: <https://arxiv.org/abs/2509.06888>. arXiv:2509.06888.
- [27] V. Basile, M. Lai, M. Sanguinetti, et al., Long-term social media data collection at the university of turin, in: Proceedings of the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), CEUR-WS, 2018, pp. 1–6.
- [28] R. van der Goot, A. Üstün, A. Ramponi, I. Sharaf, B. Plank, Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP, in: D. Gkatzia, D. Seddah (Eds.), Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Online, 2021, pp. 176–197. URL: <https://aclanthology.org/2021.eacl-demos.22/>. doi:10.18653/v1/2021.eacl-demos.22.
- [29] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
- [30] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: <http://arxiv.org/abs/1908.10084>.
- [31] C. Helwe, T. Calamai, P.-H. Paris, C. Clavel, F. Suchanek, Mafalda: A benchmark and comprehensive study of fallacy detection and classification, 2024. URL: <https://arxiv.org/abs/2311.09761>. arXiv:2311.09761.
- [32] R. Achibat, S. M. V. Hatefi, M. Dreyer, A. Jain, T. Wiegand, S. Lapuschkin, W. Samek, AttnLRP: Attention-aware layer-wise relevance propagation for transformers, in: R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, F. Berkenkamp (Eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, PMLR, 2024, pp. 135–168.

A. Additional Experimental Details

Hyper-parameters Table A.1 shows hyperparameter values for all our submitted TiGRO models. All experiments were executed on the Hábrók high-performance computing cluster at the University of Groningen, using an NVIDIA A100 GPU.

Table A.1

Training hyperparameters for TiGRO models.

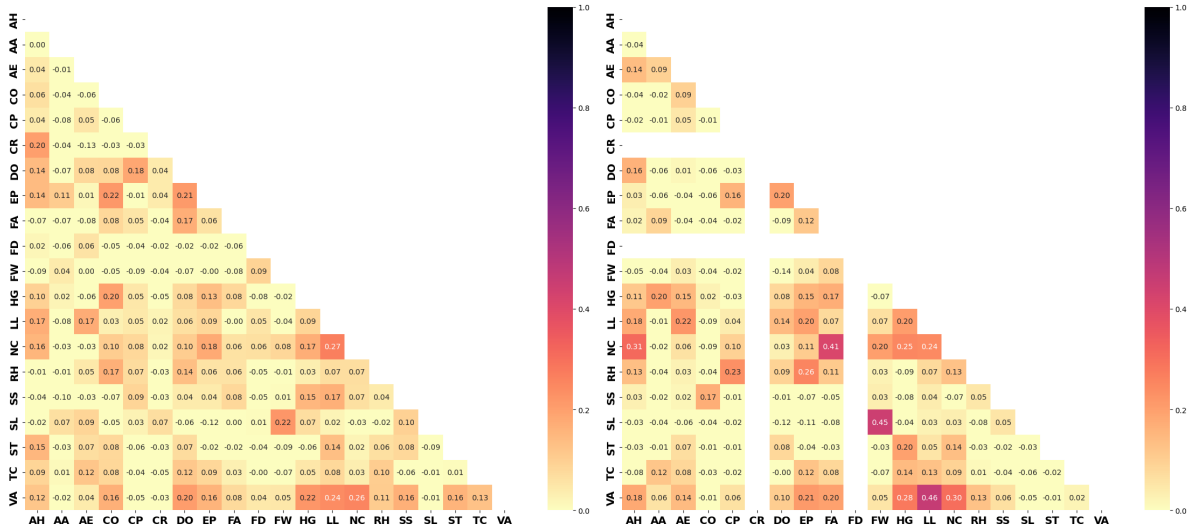
Models	Encoder	Epochs	Batch size	LR	Dropout
TiGRO A1	mmBERT	3	16	2e-5	0.0
TiGRO A2	ALBERTo	10	32	1e-4	0.2
TiGRO A3, B1, B2	ALBERTo	20	32	1e-4	0.2
TiGRO B3	mmBERT	20	32	5e-4	0.2

Subtask A Table A.2 shows TiGRO-A3 confusion matrix counts on the development set for Subtask A. Figure A.1 illustrates the correlation matrix of fallacy labels in the development set and in TiGRO-A3 prediction, showing difficulty in discerning some closely related classes, such as *Slogan* and *Flag-waving*.

Table A.2

TiGRO-A3 confusion matrix counts per fallacy type on development set, ordered by true positive count.

Fallacy (tag)	TP	FP	FN	TN
Loaded-language (LL)	121	57	11	42
Appeal-to-emotion (AE)	101	52	27	51
Vagueness (VA)	74	50	30	77
Name-calling-or-labelling (NC)	44	30	18	139
Doubt (DO)	31	9	13	178
Evading-the-burden-of-proof (EP)	15	19	29	168
Hasty-generalization (HG)	15	27	24	165
Flag-waving (FW)	12	9	15	195
Slogan (SL)	11	5	19	196
Ad-hominem (AH)	10	12	16	193
Thought-terminating-cliches (TC)	7	5	26	193
Slippery-slope (SS)	4	3	12	212
False-analogy (FA)	3	14	20	194
Appeal-to-authority (AA)	3	1	23	204
Red-herring (RH)	2	15	17	197
Causal-oversimplification (CO)	1	3	15	212
False-dilemma (FD)	0	0	7	224
Circular-reasoning (CR)	0	0	3	228
Cherry-picking (CP)	0	1	12	218
Strawman (ST)	0	2	13	216



(a) Correlation matrix for gold labels in dev-set.

(b) Correlation matrix for predicted labels (TiGRO-A3).

Figure A.1: Correlation of fallacy label co-occurrences in the development set and in TiGRO-A3 predictions, considering merged annotations.

B. Attribution Examples

The examples in Figure B.1 illustrate two true positive predictions and one false positive prediction from TiGRO-A1 for *Appeal to Emotion*, highlighting words based on attribution relevance. Specifically, red indicates a positive contribution to the predicted class, and blue indicates a negative contribution. The gold span is highlighted in **bold**.

Results are noisy, but the model generally focuses on emotion-related words supporting correct predictions for Subtask A. Example B.1a shows a perfect gold span match for Subtask B, while B.1b illustrates typical mismatches. This mainly occurs because the model assigns relevance to single key

words rather than longer spans, and is sometimes distracted by words that contribute to post-level predictions but introduce noise for exact span extraction. Example B.1c shows a false positive, where attribution can be useful for human inspection.

Decreto sicurezza : multe fino a 50mila euro e sequestro delle navi **per** le ONG che salvano migranti in mare .
Questo è un governo **disumano** , lordo e **criminale** . # decretosicurezza # governo # Meloni # criminale
(en: "Security decree: fines up to 50k euros and seizure of ships for NGOs that rescue migrants at sea. This is an *inhumane*, filthy, and criminal government. #securitydecree #government #meloni #criminal")

(a) Example of perfect match with the gold span for Subtask B.

Buongiorno a tutti , per rispetto ai tanti sfollati , alla **terribile** emergenza in Sardegna , al **disastro** ecologico , alle **vittime** innocenti animali , oggi **in segno di lutto** posterò solo informazioni riguardanti appelli urgenti e informazioni utili . Grazie Stefania
(en: "Good morning everyone, out of respect for the many displaced people, the *terrible* emergency in Sardinia, the ecological *disaster*, and the innocent animal victims, today, *as a sign of mourning*, I will only post information about urgent appeals and useful updates. Thank you, Stefania")

(b) Example of mismatch with the gold spans for Subtask B.

In 10 anni tagliati 200 ospedali , 45 mila letti , 10 mila medici e 11 mila infermieri . In 10 anni aperti 9 . 282 centri di accoglienza per immigrati . Nient ' altro da aggiungere .
(en: "In 10 years, 200 hospitals, 45,000 beds, 10,000 doctors, and 11,000 nurses have been cut. In 10 years, 9 have been opened. 282 centers for welcoming immigrants. Nothing else to add")

(c) Example of false positive, where attribution shows which textual portion has most likely led to the wrong class prediction.

Figure B.1: Attribution examples for TiGRO-A1 predictions on *Appeal to Emotion*. **Red** indicates positive contribution, **Blue** negative, and the gold span is in **bold**.

Unica at FadeIT: Adapting Large Language Models to Fallacy Identification in Social Networks

Matteo Fenu^{1,*}, Maurizio Atzori^{2,†}

¹University of Cagliari, Faculty of Science, 72 Ospedale St, Cagliari, 09124, Italy

Abstract

This paper presents a comprehensive approach to fallacy detection, addressing the challenges of handling multiple interpretations and class imbalance through a novel pre-processing pipeline. Our methodology integrates feature extraction techniques, leveraging GPT-5.1 with custom prompts to derive meaningful descriptors from social media posts, alongside data augmentation strategies combining prompt engineering and machine translation to oversample minority fallacy classes. For Subtask A, coarse-grained fallacy detection, we propose and compare three distinct approaches based on fine-tuning and Retrieval-Augmented Generation (RAG). Our experiments involve state-of-the-art large language models, including Mixtral-8x7B-Instruct, Gemma-3-12B-it, and GPT-5. Fine-tuning was performed on a single A100 GPU using the QLoRA methodology to enable parameter-efficient training. The RAG-based approach was implemented using OpenAI APIs integrated with Qdrant, an open-source vector database, for efficient semantic retrieval.

Keywords

Fallacy Detection, Large Language Models, Fine-tuning, QLoRA, Retrieval-Augmented Generation, Data Augmentation, Prompt Engineering, Feature Extraction

1. Introduction

This study contributed to the design and development of intelligent systems for the structured representation of fallacious arguments employed within social networks, through participation in sub-task A proposed by EVALITA as part of the FadeIT shared task [1]: identifying fallacies expressed in Italian social media posts. These systems aim to counteract the manipulation of public opinion and the spread of disinformation on sensitive topics in 2019-2022, such as climate change, migration, and public health.

This paper proposes two different approaches that leverage some of the most recent and high-performing foundation models, appropriately optimised through Fine-Tuning and Prompt Engineering techniques. Specifically, the first approach employs a model from the OpenAI GPT-5 family, instructed via a prompt containing dynamic few-shot examples selected on the basis of semantic similarity with the post to be classified. The second approach is based on a fine-tuning algorithm through which several Large Language Models from the Mistral, Gemma, and Llama families were developed and tested. The resulting systems achieved strong results during the challenge evaluation phase. In particular, the first approach ranked 4th among all 16 submissions with an micro F1 score of 49.71, enabling the UNICA team to secure third place in the overall standings. This paper presents an overview of the systems selected for the challenge and discusses the results obtained.

2. Motivation

The Treccani dictionary defines¹ a fallacious act, word, or argument as something deceptive: an ostensibly credible argument that is nevertheless logically flawed and therefore false. Fallacies represent

EVALITA

*Corresponding author.

†These authors contributed equally.

✉ fenum@hotmail.it (M. Fenu); atzori@unica.it (M. Atzori)

id 0009-0008-8402-9703 (M. Fenu); 0000-0001-6112-7310 (M. Atzori)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

¹<https://www.treccani.it/vocabolario/fallacia/>

a significant risk to public safety today, particularly in a society constantly exposed to the latest news or the most recent informative post published on social networks.

Fallacies are also a tool widely exploited by fraudsters to deceive individuals and prompt them to action. An emblematic case is so-called phishing, namely the attempt to illegally obtain a user's personal data, along with other sensitive information, for the purpose of theft. This practice, which particularly affects elderly individuals, relies on the dissemination of deceptive messages constructed through fallacies, with the aim of extracting the desired information. This constitutes a substantial risk and a matter of paramount importance in the ongoing effort to protect personal data.

The research aims to leverage the challenge proposed by the EVALITA 2026 campaign in order to develop tools useful to the public for counteracting fake news, the manipulation of information and opinions, as well as fraud attempts and the misappropriation of personal data. From a technical standpoint, the objective of this research is to assess how the most recent Large Language Models can be adapted to perform multi-class tasks such as that proposed for sub-task A (detection of fallacies in social media posts), and to evaluate the impact of model size and the optimisation algorithms selected.

3. Definition of the tasks and evaluation measures

The Subtask A proposed by the organisers, the one on which this research focuses, is defined as follows: given a text, predict the set of fallacies expressed within it. This is a multi-label classification task. Systems are evaluated using Precision, Recall, and micro F1-score, computed as the average across the two annotations (both considered equally valid). The metric that determines the final ranking is the micro F1-score. For informational purposes only, and not covered in this study, Subtask B is defined as follows: given a text, predict the segments containing fallacies. This is a multi-label sequence labelling task. The main difficulty lies in the fact that fallacies may partially overlap with one another. In this case, the text is represented as a sequence of tokens. The annotation follows the BIO scheme (B: begin, I: inside, O: outside); any multiple annotations for the same token are separated by the pipe symbol (|) and preceded by the corresponding tag. The metrics adopted are variants of F1-score, Precision, and Recall, adapted to operate at the token level. The evaluation is defined as soft, as it also accounts for partial matches, weighted proportionally to the length of the match itself in terms of tokens.

Table 1
Example of annotation for Subtask A

Text	Annotator 1		Annotator 2	
Basta con le politiche immigrazioniste della sinistra. Il #25settembre vota [USER] [URL]	Flag-waving,	Name-calling-or-labelling, Slogan	Flag-waving,	Loaded-language, Name-calling-or-labelling, Slogan

English translation: “Enough with the left’s pro-immigration policies. On #September25 vote [USER] [URL]”

4. Dataset overview

The entire FadeIT shared task is based on the Faina dataset [2], a dataset for fallacy recognition. It contains social media posts written in Italian. The main topics addressed within it concern immigration, climate change, and public health.

The dataset is based on an ontology of twenty fallacy types, such as, for example, appeals to authority or to readers’ emotions, hasty generalisations drawn from small and unrepresentative samples, or the use of slogans or phrases intended to elicit enthusiasm among the readership.

The dataset provides two different types of annotations, from which the two different sub-tasks proposed for the challenge are derived:

- train-dev.tsv: Each element of the dataset is characterised by the post text, the topic of the post, and two annotations relating to the fallacies present. Each annotation contains a list of fallacies separated by the pipe symbol (|). This is the dataset proposed for sub-task A.
- train-dev.conll: Compared to the .tsv file, the annotations in this dataset contain a tag for each element of the sentence (token) using the BIO scheme (Begin, Inside, Out). This is the dataset proposed for sub-task B.

From an initial analysis phase of the available dataset, two main criticalities emerged. The first concerns a suboptimal distribution of the number of examples for each fallacy. Indeed, within the dataset, certain fallacies predominate with over 10% coverage (loaded-language, appeal-to-emotion), while others do not exceed 2% (circular-reasoning, false-dilemma, cherry-picking). The second, more subtle in nature and certainly more difficult to address, concerns the lack of immediate comprehension of the meaning of certain fallacies for a transformer-based model. Fallacies such as circular reasoning, strawman, cherry picking, and false analogy prove particularly challenging at first approach and require structured reasoning capabilities to be properly understood.

Overall, the dataset comprises 1152 posts. Of these, 107 contain no fallacies according to both annotators. On average, each post contains 2.79 fallacies according to the first annotator and 2.96 according to the second annotator. Another noteworthy aspect is the average length of spans containing a given fallacy. Some fallacies, such as Loaded-language or Name-calling-or-labelling, are characterized by approximately two words per span on average. Among the most complex in this regard are Evading-the-burden-of-proof with 16.47 average words per span and Circular-reasoning with 28.69. The Avg. Words column in Table 2 summarizes the values for each fallacy. Additionally, Table 2 reports the number of posts in which each fallacy occurs and the corresponding percentage relative to the total number of posts in the dataset.

Table 2
Distribution of Fallacies in the original training set (Part 1)

Fallacy	Occurrences	Percentage (%)	Avg. Words
Loaded-language	1158	17.49	2.54
Appeal-to-emotion	1103	16.66	5.08
Vagueness	816	12.32	9.24
Name-calling-or-labelling	603	9.11	2.61
Doubt	366	5.53	16.30
Hasty-generalization	344	5.20	10.50
Evading-the-burden-of-proof	294	4.44	16.47
Flag-waving	248	3.75	4.28
Ad-hominem	234	3.53	15.57
Slogan	226	3.41	3.45
Thought-terminating-cliches	214	3.23	5.26
Red-herring	187	2.82	12.51
False-analogy	186	2.81	21.36
Appeal-to-authority	159	2.40	6.15
Slippery-slope	135	2.04	10.48
Causal-oversimplification	108	1.63	19.51
Strawman	87	1.31	37.07
Cherry-picking	76	1.15	29.18
False-dilemma	61	0.92	16.31
Circular-reasoning	16	0.24	28.69

4.1. Pre-processing

To prepare the dataset for the subsequent training and evaluation phases, a series of operations were performed aimed at increasing its quality and comprehensibility, bearing in mind the criticalities

encountered during the initial analysis phase.

4.1.1. Data Augmentation

First, we performed a data augmentation operation in order to reduce the gap between the less representative fallacies and the more frequent ones. Specifically, these operations focused on increasing the number of examples for fallacies that exhibit poor representation in the dataset. To achieve this objective, two different strategies were adopted.

- Prompt engineering: a prompt was developed to instruct GPT 5.1 to generate new examples for the set of minority fallacies. As can be seen from Figure 1, in the first static part of the prompt, the task to be performed and the structure of the expected output are defined. The second part, which varies according to the fallacy for which examples should be augmented, contains the fallacy name, its description in natural language, and a list of example social media posts containing it drawn from the dataset. The algorithm was executed on the set of fallacies with a percentage below 3% in the original training set. For each of these, fifty new examples were generated for inclusion in the final dataset to be used for system training.

System Prompt: Fallacy Generator

Act as a creator of posts containing fallacies. Your task is to generate new comments containing the target fallacy. Create fallacies containing no more than 40 words.

Output format: JSON

```
{
  "examples": [
    "comment with target fallacy",
    "comment with target fallacy",
    ...
  ]
}
```

Input variables:

- <target_fallacy>: The fallacy for which to generate new examples
- <fallacy_description>: Description of the target fallacy
- <list_of_examples>: Examples to draw inspiration from

Figure 1: Prompt template for fallacy generation via LLM

- Back-translation: this is the process of retranslating content from the target language back to the source language in literal terms. During this phase, semantically equivalent paraphrases of existing posts were generated through the use of pivot translations. This process involves translating the original text into an intermediate language and subsequently retranslating it into the original language (Italian). Two translation cycles were employed: Italian-Spanish-Italian and Italian-French-Italian. Prior to performing the actual translation, hashtags were masked using placeholder tokens to prevent them from being altered. The Helsinki-NLP neural models from the OPUS-MT family were used for translation. Specifically, we employed: opus-mt-it-es and opus-mt-es-it for Italian-Spanish translation, and opus-mt-it-fr and opus-tatoeba-fr-it for Italian-French translation. [3] Once the paraphrase was obtained, a quality filter based on semantic similarity was applied. The embeddings of the post before and after the back-translation process were computed, and the cosine similarity between the two resulting vectors was calculated. Paraphrases with a similarity below 75% were discarded. In the final step, the placeholders inserted in the first phase were restored with their original values. During this phase, as the number of examples generated with the first approach for minority classes was deemed sufficient, the focus was not placed on particular classes; rather, a random selection of rows to be paraphrased was performed in order to increase the overall quantity of examples in the training dataset.

The aforementioned operations led to the creation of the final dataset ready for model training. The final number of rows is 2973, compared to the initial 1152. The resulting dataset shows a substantial increase in the number of examples for several classes, including Slippery-slope, Strawman, and Cherry-picking, thereby leading to greater representativeness for all target fallacies.

4.1.2. Features extraction

The second fundamental step of the pre-processing phase is feature extraction. The objective is to extract useful information and characteristics from each post to increase the likelihood of correctly classifying fallacies. In the first phase, the posts and their corresponding fallacy annotations were related to the natural language descriptions of the fallacies themselves. In this way, it was possible to identify certain fundamental characteristics to support classification. This step was performed manually without algorithmic support. To this end, the posts and their corresponding fallacy classifications were analyzed individually. The classifications were then related to the definition of each fallacy in order to identify common characteristics across classifications useful for simplifying the classification task for the model. Specifically, the features extracted in this phase are the following: `doubt_raising` (boolean), `unsupported_claim` (boolean), `analogy` (boolean), `flag_waving` (boolean), `generalization` (boolean), `profanity` (boolean), `thought_terminating` (boolean), `labeling` (list), `red_herring` (list), `slogans` (list), and `vague_terms` (list). Boolean features indicate the presence or absence of a specific linguistic pattern, while list features contain the extracted text spans matching that pattern. In total, 11 features are extracted for each post. The subsequent step consists of creating a structured prompt containing a clear definition of the problem (identification of the features characterising each fallacy) and an explanation of the expected output format. Subsequently, the prompt is used to instruct GPT-5 configured with enhanced reasoning capabilities in order to analyse the post, extract the relevant features, and save them in a dedicated JSON file. Some of the extracted features concern appeals to patriotism, tautological reasoning, discursive simplifications, or the use of ambiguous terms. The output of this phase was subsequently used in the first approach as additional information aimed at better describing the post to be classified and the corresponding example posts.

```
"doubt_raising": boolean,
"unsupported_claim": boolean,
"analogy": boolean,  "flag_waving": boolean,
"generalization": boolean,
"profanity": boolean,
"labeling": [],
"red_herring": [],
"slogans": [],
"thought_terminating": boolean,
"vague_terms": []
} Input: {comment} Output: Valid JSON only. No explanations.
```

Figure 3: Prompt template for rhetorical feature extraction via LLM (Part 2)

System Prompt: Features extractor

*Act as a rhetorical feature detector. Your task is to analyze a social media comment and return a JSON object detecting the following persuasion/manipulation techniques. **Output format:** JSON {*

```
"personal_attack": boolean,  
"authority": [],  
"emotional_appeal": boolean,  
"oversimplification": boolean,  
"tautology": boolean,
```

Figure 2: Prompt template for rhetorical feature extraction via LLM (Part 1)

4.2. First approach - Large Language Models Optimised through RAG and Dynamic Few-Shot Prompting

For the first approach to the fallacy identification problem, we chose to employ a prompt enriched with examples using the Retrieval Augmented Generation (RAG) technique applied to a vector database containing the training dataset. The RAG technique is used to retrieve information from an external database in order to enhance the quality of responses generated by the model.

The entire system was developed with the aid of the agentic orchestration library by DataPizza. DataPizza is an Italian open-source project developed to address the limitations of existing traditional agentic frameworks. Its strengths include minimal abstraction, a debugging-oriented workflow, maximum flexibility, and customisation through the use of components that can be tailored to developers' requirements.²

As with a traditional vanilla RAG system, the post classification system is composed of a retriever, responsible for retrieving information from the external memory source, and a fallacy generator instructed via prompt. [4]

4.2.1. Retriever

The underlying idea is to use the post to be classified as a search query on the vector database to retrieve examples of similar posts and their corresponding classifications.

For implementation, the advanced open-source vector similarity search technology from Qdrant was employed. It offers a wide range of functionalities, including the ability to implement vector databases and similarity search algorithms. The collection configuration for test set data storage is as follows:

- **Embedding model:** text-embedding-3-small model from the OpenAI family;
- **Dimension:** the dimension corresponds to the dimensionality of the vectors generated by the model, namely 1536;
- **Format:** dense vectors are used;
- **Distance:** the metric for calculating similarity between the query and the vectors is cosine measure, which calculates the angle between vectors in the space.

As a preliminary step, we partitioned the original training set using a 70/30 split, allocating 70% of the data for training and the remaining 30% for validation. The first step in development is represented by the ingestion pipeline. During this phase, each post belonging to the training set was pre-processed, transformed into embeddings, and saved within the collection on Qdrant. Along with the embeddings, for each element, the features extracted during pre-processing, the fallacy labels assigned by annotators A and B, and the token-level classification from the .conll file were stored as metadata. For the retrieval phase, the post to be classified is converted into vector format and the top five most similar examples are retrieved using cosine measure.

²<https://datapizza.tech/en/ai-framework/>

4.2.2. Generator

The model family used in this phase is that of OpenAI. Specifically, tests were conducted on GPT-4.1 and GPT-5.

The second step consists of defining the prompt, containing all the instructions necessary to train the model. The prompt is composed of the following elements:

1. Problem definition
2. Description of the fallacy ontology: for each fallacy, a textual description is provided in order to give the model greater context.
3. Output definition: The expected output is a JSON containing a list of Prediction objects (sub-text, fallacy, confidence). During the post-processing phase, predictions with confidence level below 60% are discarded.
4. Suggestions to avoid errors: certain erroneous behaviours were identified during the development phase. Specific instructions were inserted to circumvent such issues, whilst avoiding leading the model into overfitting with overly specific indications.
5. List of similar examples: cosine similarity was used as the measure for identifying similar elements. The post to be classified is transformed into embeddings and the three most similar elements are identified. These elements, together with their respective classifications, are provided to the model within the prompt to guide it towards correct classification. In addition to these elements, the features extracted during the feature extraction phase are also included in the prompt. This was done to simplify the model's reasoning and guide it in establishing relationships between the features of the post to be classified and those of the posts suggested as similar examples. A possible future development for improving the system is represented by the use of features for retrieving similar examples, through their conversion into embeddings and the combination with the embeddings of the various posts.
6. Features of the post to be classified: using the feature extraction algorithm described above, the features of the post to be classified were extracted and included in the prompt as additional information.

System Prompt: Fallacy Classification

You are an annotator for comments containing fallacies. Given a social media post, identify the fallacies expressed in it. This is a multi-label classification task using 20 fallacy classes: Ad hominem, Appeal to authority, Appeal to emotion, Causal oversimplification, Cherry picking, Circular reasoning, Doubt, Evading the burden of proof, False analogy, False dilemma, Flag waving, Hasty generalization, Loaded language, Name calling or labeling, Red herring, Slippery slope, Slogan, Strawman, Thought-terminating cliché, Vagueness.

Input:

- List of semantically similar examples
- Comment to predict

Output format: JSON

```
[  
{ "text": "sub-phrase", "fallacy": "prediction", "confidence": "0-100%"},  
...  
]
```

Guidelines:

- Return only elements in the required JSON format
- Discard predictions with confidence below 60%

Figure 4: Prompt template for the first approach

4.3. Experiments

Tests were conducted on a Mac Mini M4, using the models via the OpenAI APIs. During the system evaluation phase, two different versions were analyzed. The first involved a simplified version of the output object returned by the model, without confidence scores and containing only the list of fallacies for each similar example included in the prompt. The second version, subsequently adopted during the final evaluation phases on the test set, is characterized by the use of confidence scores for each prediction, in order to assess how reliable the model considers a given prediction. Furthermore, this version also includes span-level classification (extracted from the .conll file) to show, for each example, the specific portions in which the fallacies in the training set were identified by the annotators.

Table 3

Model Experiments on the development set

Model	Precision	Recall	Micro F1-Score
GPT 5.1 simplified version	0.54	0.50	0.52
GPT 5.1 final version	0.61	0.56	0.56

4.4. Second approach - Large Language Models Optimised through Supervised Fine-Tuning

This section describes the supervised fine-tuning algorithm used to optimise pre-trained open-source models on the classification task. The results of runs 2 and 3 are subsequently derived from these models.

This algorithm is based on QLoRA (Quantized Low-Rank Adaptation) techniques and leverages the Hugging Face Transformers (4.56.1) and PEFT-Parameter-Efficient Fine-Tuning (0.18.0) libraries.

In summary, the algorithm is structured as follows:

- **Data preparation and pre-processing:** The data preparation and pre-processing phase begins with the definition of the list of target labels, namely the fallacies to be predicted. Subsequently, the reference dataset is loaded, pre-processed, and split into training set (70) and development set (30). The fallacy labels are then converted from textual format to one-hot encoding format, creating a binary column (values 0 or 1) for each fallacy. Furthermore, label weights are calculated to mitigate class imbalance during training. The subsequent step involves tokenising the posts and creating training batches through padding. Finally, the fallacy labels are converted into PyTorch tensors.
- **Model configuration and PEFT:** This phase begins with the definition of quantisation via the BitsAndBytes library (0.49.1). 4-bit quantisation is selected to implement the QLoRA (Quantized Low-Rank Adaptation) technique. This choice significantly reduces VRAM memory usage, enabling the fine-tuning of large-scale models on a single A100 GPU. Subsequently, the LoRA configuration is defined by specifying its parameters, particularly the target_modules (qproj, kproj, vproj, oproj), which represent the layers to be adapted. Finally, the model is loaded, quantised, and configured with LoRA.
- **Training:** This crucial phase defines the hyperparameters and initiates the model fine-tuning process. The main hyperparameters include:
 1. Learning rate: 10^{-4} ;
 2. Batch size per device: 4;
 3. Evaluation and saving strategy: steps;
 4. Load best model at end: true.

Subsequently, a CustomTrainer (Python class) is initialised. This trainer is fundamental as it enables the management of a custom loss function that utilises the label weights calculated

previously. The use of weights serves to balance the impact of less frequent classes (rare fallacies) during model optimisation.

Finally, the training process is initiated and, upon completion, predictions on the validation set are computed and performance metrics are recorded, preparing the model for the subsequent evaluation phase on the test set.

4.4.1. Experiments

The following models were trained: gemma-3-12b-it, Meta-Llama-3.1-70B-Instruct, and Mixtral-8x7B-Instruct-v0.1. The fine-tuning process was carried out on a device with 235GB of storage and an A100 graphics card with 80GB. Once training was completed, the models were evaluated on the development set extracted from the training dataset. The model with the best result in terms of micro F1-score is Mixtral, followed by Gemma. For this reason, they were selected for the final submission of predictions on the official test set. Run 2 corresponds to the result of Gemma, whilst run 3 corresponds to that of Mixtral.

Table 4
Model Experiments on the development set

Model	Precision	Recall	Micro F1-Score
Mixtral-8x7B-Instruct-v0.1	0.58	0.71	0.64
Gemma-3-12b-it	0.53	0.71	0.61
Meta-Llama-3.1-70B-Instruct	0.74	0.47	0.57

5. Results

Table 5 summarises the final results of the predictions on the test set provided by the organisers of the FadeIT task. The test set portion of the dataset consists of 288 social media posts with a substantial balance in the topics addressed therein (climate change, public health, and immigration).

Table 5
Subtask A results

	Pos.	avg			a1			a2		
		P	R	F1	P	R	F1	P	R	F1
GPT-5	4/16	55.22	45.23	49.71	53.62	45.78	49.39	56.82	44.69	50.03
Mixtral-8x7B-Instruct-v0.1	6/16	51.19	44.26	47.45	48.09	43.40	45.63	54.28	45.13	49.28
Gemma-3-12b-it	8/16	58.70	38.44	46.44	54.78	37.46	44.49	62.61	39.43	48.39

6. Discussion

The approach that produced the best results is few-shot prompting via RAG. The fourth position obtained secured third place as a team in the final ranking.

Gemma and Mixtral, on the other hand, ranked further behind. In particular, Gemma-3, despite achieving higher precision compared to the other models, recorded the lowest score in terms of micro F1-score. Overall, the final scores below 50% attest to the intrinsic difficulty of the classification task, considering the very high number of classes involved (20). Another interesting aspect to highlight concerns the higher scores for annotator two, particularly with regard to the models optimised through fine-tuning.

The final results also demonstrate how the initially superior performance of the models produced through fine-tuning subsequently proved worse on the test set. It will be interesting to evaluate this aspect with respect to the distribution of fallacies in the training and test datasets.

7. Related works

The automatic recognition of fallacies and propaganda techniques has received increasing attention in recent years. SemEval, for instance, has introduced this topic in numerous tasks such as SemEval-2020 Task 11: Detection of Propaganda Techniques in News Articles [5] and SemEval-2021 Task 6: Detection of Persuasion Techniques in Texts and Images [6].

This study takes as its starting point the research conducted within SemEval-2025 Task 10 [7], which focused on the automatic identification of narratives, their classification, and the determination of the roles of entities involved in a dataset of news articles. This experimentation focused on the fine-tuning of Llama 3 on sub-task 1 of Entity Framing. The resulting system achieved third place for the English language dataset and led to the production of a scientific paper [8]. The promising results of this research constituted the foundation for the development of the second approach (Large Language Models Optimised through Supervised Fine-Tuning).

8. Conclusion

This study allowed for the implementation and evaluation of the performance of current foundation models available on the market on the task of classifying Italian social media posts. In particular, the best results were achieved through the RAG technique and few-shot prompting, leveraging the most performant OpenAI models available on the market. Through this approach, the final system produced achieved the highest score in terms of micro F1-score among the three runs submitted by the UNICA team for evaluation. The scores in the final ranking attest to the overall difficulty of Task A proposed by the organizers, thus leaving room for further research and analysis in this domain.

Acknowledgments

This work has been partially supported by the projects DEMON “Detect and Evaluate Manipulation of ONline information” funded by MIUR under the PRIN 2022 grant 2022BAXSPY (CUP F53D23004270006, NextGenerationEU) and SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU (NextGenerationEU (CUP F53C22000740007)).

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Opus 4.5 in order to: Grammar and spelling check and Text Translation. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication’s content.

References

- [1] A. Ramponi, S. Tonelli, FadeIT at EVALITA 2026: Overview of the fallacy detection in italian social media texts task, in: Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026), CEUR.org, Bari, Italy, 2026.
- [2] A. Ramponi, A. Daffara, S. Tonelli, Fine-grained fallacy detection with human label variation, in: L. Chiruzzo, A. Ritter, L. Wang (Eds.), Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language

- Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Albuquerque, New Mexico, 2025, pp. 762–784. URL: <https://aclanthology.org/2025.naacl-long.34/>. doi:10.18653/v1/2025.naacl-long.34.
- [3] J. Tiedemann, S. Thottingal, OPUS-MT – building open translation services for the world, in: A. Martins, H. Moniz, S. Fumega, B. Martins, F. Batista, L. Coheur, C. Parra, I. Trancoso, M. Turchi, A. Bisazza, J. Moorkens, A. Guerberoef, M. Nurminen, L. Marg, M. L. Forcada (Eds.), Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, European Association for Machine Translation, Lisboa, Portugal, 2020, pp. 479–480. URL: <https://aclanthology.org/2020.eamt-1.61/>.
- [4] C. Huyen, AI Engineering: Building Applications with Foundation Models, O’Reilly Media, 2024. URL: <https://www.oreilly.com/library/view/ai-engineering/9781098166298/>.
- [5] G. Da San Martino, A. Barrón-Cedeño, H. Wachsmuth, R. Petrov, P. Nakov, SemEval-2020 task 11: Detection of propaganda techniques in news articles, in: A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, E. Shutova (Eds.), Proceedings of the Fourteenth Workshop on Semantic Evaluation, International Committee for Computational Linguistics, Barcelona (online), 2020, pp. 1377–1414. URL: <https://aclanthology.org/2020.semeval-1.186/>. doi:10.18653/v1/2020.semeval-1.186.
- [6] D. Dimitrov, B. Bin Ali, S. Shaar, F. Alam, F. Silvestri, H. Firooz, P. Nakov, G. Da San Martino, SemEval-2021 task 6: Detection of persuasion techniques in texts and images, in: A. Palmer, N. Schneider, N. Schluter, G. Emerson, A. Herbelot, X. Zhu (Eds.), Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), Association for Computational Linguistics, Online, 2021, pp. 70–98. URL: <https://aclanthology.org/2021.semeval-1.7/>. doi:10.18653/v1/2021.semeval-1.7.
- [7] J. Piskorski, T. Mahmoud, N. Nikolaidis, R. Campos, A. Mario Jorge, D. Dimitrov, P. Silvano, R. Yangarber, S. Sharma, T. Chakraborty, N. Guimaraes, E. Sartori, N. Stefanovitch, Z. Xie, P. Nakov, G. Da San Martino, SemEval 2025 task 10: Multilingual characterization and extraction of narratives from online news, in: S. Rosenthal, A. Rosá, D. Ghosh, M. Zampieri (Eds.), Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 2610–2643. URL: <https://aclanthology.org/2025.semeval-1.331/>.
- [8] M. Fenu, M. Sanguinetti, M. Atzori, DEMON at SemEval-2025 task 10: Fine-tuning LLaMA-3 for multilingual entity framing, in: S. Rosenthal, A. Rosá, D. Ghosh, M. Zampieri (Eds.), Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, Vienna, Austria, 2025, pp. 1456–1464. URL: <https://aclanthology.org/2025.semeval-1.192/>.

A. Online Resources

The code of the work conducted is available at the following link:

- <https://github.com/Matteofenu65891/task-A—FadeIt.git>

For clarity of reading, all prompts described in this paper have been translated into English. The original Italian versions are available in the repository.

IMPOLS at EVALITA 2026: Overview of the IMPOLS Task^{*}

Lorenzo Gregori^{*,†}, Walter Paci[†] and Valentina Saccone[†]

University of Florence

Abstract

This paper describes the IMPOLS task presented at EVALITA 2026 competition. The task regards the automatic detection and classification of implicit contents in political speeches, specifically focusing on the uses of implicit expressions that are potentially manipulative. The task is divided in three subtasks: (1) implicit content detection, (2) implicit classification (implicature and presupposition), and (3) implicature type classification. IMPOLS is monolingual (Italian) multimodal task: provided data included audio and transcription of speech excerpts, along with some discourse metadata. Two teams participated at IMPOLS, obtaining a macro F1 score of 0.95 on subtask 1, 0.88 on subtask 2, and 0.64 on subtask 3.

Keywords

political speech, implicit content, EVALITA 2026,

1. Introduction and motivations

Political communication systematically relies on meanings that are not explicitly asserted but still conveyed and taken for granted by the audience. Presuppositions and implicatures allow speakers to introduce contested propositions as shared background knowledge or to suggest evaluative content while limiting explicit commitment. In political contexts, these mechanisms are not incidental rhetorical effects but instruments for persuasion, accountability management, and framing [1, 2, 3].

Implicit meanings that are understood as true within a given context, but that are conveyed strategically rather than cooperatively, often with manipulative intent, are called **non-bona fide true**. As shown in recent pragmatic and computational studies, these contents exploit the inferential dimension of language to bypass active judgment, making them difficult to detect both for humans and automatic systems [4, 5, 6]. To illustrate the point discussed above, we present an example of a non-bona fide true implicit content (in *italics*), with its explicit explanation.

È stato già preparato dall'associazione pescatori, *giace come tanti altri piani nei cassetti della Regione.*
"It has already been prepared by the fishermen's association, it is sitting, like many other plans, in the drawers of the Regional Government."
Debora Serracchiani 2013

This excerpt leads the hearer to imply that, given the presence of many other unconsidered plans, the current Regional Government does not really address the fishermen's problems. Thus, the speaker, while talking about a new plan, stresses the current Government inadequacy without explicitly saying it and, for this reason, without taking responsibility of this implicit judgment.

From a computational perspective, the automatic identification of implicit content poses challenges that go beyond surface-level semantic classification: implicit meanings depend on contextual reasoning, shared knowledge, discourse structure, and pragmatic expectations, all of which are only partially encoded in the linguistic signal [1, 2].

The IMPOLS task (**IM**PLICIT contents in **POL**itical **S**peech) at EVALITA2026 [7] is designed to address this challenge by providing a structured evaluation framework for the automatic detection and classification of non-bona fide true implicit content in political discourse. The task builds on the IMPAQTS

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

^{*}Corresponding author.

[†]These authors contributed equally.

✉ lorenzo.gregori@unifi.it (L. Gregori); walter.paci@unifi.it (W. Paci); valentina.saccone@unifi.it (V. Saccone)

ORCID 0000-0001-9208-2311 (L. Gregori); 0009-0000-9804-7699 (W. Paci); 0000-0002-6222-6326 (V. Saccone)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

corpus, a large-scale, manually annotated collection of Italian political speeches spanning the period from 1946 to 2023, specifically developed to capture pragmatic implicitness in naturalistic, ecological communicative contexts [5, 6]. Unlike prior work, which relies on English synthetic stimuli or isolated sentences, IMPOLS focuses on semi-spontaneous political speech and preserves its multimodal and contextual properties.

IMPOLS is articulated into three subtasks that reflect increasing levels of pragmatic specificity: (i) the detection of questionable implicit content, (ii) the classification of such content into implicatures and presuppositions, and (iii) the fine-grained classification of implicature types. This task decomposition is theoretically motivated by pragmatic theory and empirically justified by previous findings showing that different implicit phenomena pose distinct computational challenges and error profiles [1, 5].

IMPOLS aims to promote the development of models that go beyond lexical cues and shallow correlations, and to foster research on interpretable, context-sensitive approaches to political language analysis. The task contributes to the growing effort to assess whether current NLP systems can meaningfully engage with the inferential dimension of language that underlies persuasive and manipulative communication [3, 4].

2. Dataset

The dataset used for this task is derived from the IMPAQTS corpus¹, a large-scale, multimodal collection of Italian political speeches systematically annotated for pragmatic implicitness [5, 6]. The IMPAQTS corpus is a comprehensive resources for studying Italian political discourse: it includes speeches from prominent Italian political figures, spanning the entire period of the Italian Republic from 1946 to 2023. Each speech is manually annotated for various forms of non-explicit persuasion, including presuppositions, implicatures, vague expressions, and topicalizations². Speech transcriptions are manually revised and time-aligned to the original audio/video source, making IMPAQTS a fully multimodal resource for the analysis of political speech.

The annotation of IMPAQTS has been performed in parallel by 3 linguists with a strong expertise on implicit communication. These individual annotation has been merged by an expert reviewer, who took the final decision in the cases of full disagreement, i.e. whenever there is not an agreement of 2/3 on the presence or the type of the implicit content.

The dataset used for this task contains 2,637 excerpts from IMPAQTS, divided in this way: 1,332 excerpts with questionable implicit contents and 1,305 without them; half of the excerpts with implicit contents (666) contain implicatures and the other half contain presuppositions; in turn, implicatures are subdivided into three classes: particularized conversational, generalized conversational, and conventional implicatures, each one consisting of 222 elements.

Data are extracted from public speeches freely available on YouTube. Labels used in the task refer exclusively to implicit contents that are not *bona-fide* true, namely, persuasive and manipulative linguistic strategies. For each excerpt, we provide the speech (as an audio file), the manually revised transcription, and the speech metadata.

The dataset is available on the IMPOLS task website³.

2.1. Data format

IMPOLS dataset is released in tab-separated CSV format. The format is the same for each subtask and contains the following columns:

- *id*: the unique id of the sentence;
- *speaker*: the name of the speaker;
- *year*: the year the speech was delivered;

¹<https://impaqts.dilef.unifi.it/>

²Examples of implicit types can be found in Appendix B.

³<https://liu-dilef.github.io/IMPOLS-task/>

- *text*: the speech transcription, containing the text to classify within an <s> tag and a wide context;
- *tag*: the annotation tag, that is
 - {0,1} for subtask 1: the segment within <s> contains a *bona-fide* true implicit (1) or not (0);
 - {IMP,PPP} for subtask 2: the segment within <s> contains an implicature (IMP) or a presupposition (PPP);
 - {IMPcvrs,IMPconv,IMPgen} for subtask 3: the segment within <s> contains a particularized conversational implicature (IMPcvrs), a generalized conversational implicature (IMPgen), or a conventional implicature (IMPconv).

An audio file in MP3 format is provided for each sentence and is named *[id].mp3*.

3. Tasks

IMPOLS is divided in three subtasks:

- a binary detection task, in which the systems are asked to detect the presence of questionable implicit contents in speech excerpts;
- a binary classification task, in which the systems are asked to discriminate between two types of implicit contents: implicatures and presuppositions;
- a multiclass classification task, in which the systems have to identify if implicatures are particularized conversational, generalized conversational, or conventional.

3.1. Subtask 1. Implicit content detection

The implicit content detection task consists of identifying the presence of questionable implicit content in speech excerpts. It is applied to the full dataset of 2,637 excerpts split into training and test sets. Predictions will be evaluated through the F1-score.

classes	train	test	total
implicit	912	420	1332
no_implicit	912	393	1305
total	1824	813	2637

Table 1

Subtask 1 dataset numbers.

3.2. Subtask 2. Implicit classification

The implicatures classification task involves a binary distinction between implicature and presupposition. These two labels can be briefly defined as follows:

- Implicature refers to the mechanism through which a meaning not explicitly stated or expressed is suggested. As the content is inferred by the recipients themselves, they are often less aware that it has been conveyed to them, and they are less likely to question it. As an instance of content implicitness, implicatures induce the addressee to extract further, unexpressed meanings from what is said.
- Presupposition refers to the mechanism by which a piece of information is presented as if the recipients were already familiar with it. Because it is framed as shared knowledge, recipients are led to take it for granted and are therefore less likely to critically evaluate it. As an instance of responsibility implicitness, presuppositions attribute responsibility for the content also to the addressee.

This subtask is applied to the subset of 1332 excerpts containing implicit content and is split into training and test sets. Predictions are evaluated through the F1-score.

classes	train	test	total
implicatures	456	210	666
presuppositions	456	210	666
total	912	420	1332

Table 2

Subtask 2 dataset numbers.

3.3. Subtask 3. Implicatures classification

The implicatures classification task is applied to a subset of 666 excerpts, which must be categorized into one of three types: particularized conversational, generalized conversational, or conventional implicatures.

- **Particularized Conversational Implicatures:** In line with Grice’s theoretical framework, conversational implicatures are defined as those arising when the speaker deliberately (or seemingly deliberately) challenges one of the four conversational maxims derived from the Cooperative Principle. More specifically, in this type of implicature, the intended inference depends on particular features of the specific context of the utterance, which occur as one-off inferences and are not generalizable across different situations.

Storicamente le donne sono state fuori del potere perché non partecipavano. <s>Perciò noi non abbiamo il retaggio di un potere che ci abbia intaccato, corrotto</s>.

"Historically speaking, women were excluded from power because they did not take part in it. <s>Therefore, we do not carry the legacy of a form of power that may have tainted, corrupted us</s>."

Tina Anselmi, 1991

- **Generalized Conversational Implicatures:** Like the previous category, these generalized implicatures are conversational. For these, the hearer assumes that the speaker is obeying the maxims; they only depend on what the speaker has said and on general assumptions regarding cooperative communication. They do not require specific contextual knowledge for interpretation.

Oggi ci siamo, <s>noi non scappiamo di fronte alle difficoltà</s>.

"Today we are here, <s>we do not run away from difficulties</s>."

Francesco Storace, 2011

- **Conventional Implicatures:** They do not depend on the context of utterance (e.g., communicative situation, conversational goals, or adherence to the Cooperative Principle) but are rather stably associated with the conventional meaning of certain expressions. Their interpretation requires knowledge of the lexical or grammatical item from which they arise.

Ma <s>neanche su questo noi abbiamo sentito una seria affermazione [...]</s>.

"But <s>not even on this matter have we heard any serious statement [...]</s>."

Pino Rauti, 1978

Data are split into training and test sets. Predictions will be evaluated through the macro F1-score.

classes	train	test	total
particularized conversational impl.	152	70	222
generalized conversational impl.	152	70	222
conventional impl.	152	70	222
total	456	210	666

Table 3

Subtask 3 dataset numbers.

4. Participants and results

Two teams participated in IMPOLS task: *RES2* [8] and *kenji-endo* [9]. In particular, *RES2* participated in all the 3 subtasks and submitted 2 runs; *kenji-endo* participated in subtask 1 and 2 and submitted 1 run.

4.1. Developed systems

RES2 created a system that exploits UmBERTo for semantic encoding and a multilayer perceptron as a classifier. The system uses the concatenation a global vector, encoding the semantic representation of the whole text, with a local one, that represents only the text span to classify. Interestingly, a data augmentation strategy based on Gemini has been employed to increase the training dataset. *RES2* created a text-only system, that does not consider audio and metadata to perform classification. The system has been used for all the 3 subtasks.

kenji-endo trained a BabyLM from scratch and exploited it to participate in four EVALITA tasks, including IMPOLS. The model has been pre-trained on a curated corpus of about 54M tokens and has approximately 130M parameters. The model variant named *Kenji-Endo vanilla* has been used to address IMPOLS subtasks 1 and 2: for the detection task, the model has been fine-tuned on the training set, while for the implicit classification the tuning is performed through prompt refinement. As the system developed by *RES2*, *kenji-endo* did not exploit multimodal features, nor discourse metadata.

4.2. Results

Table 4 shows the results of macro-averaged F1 score obtained on the three subtasks by the two teams. The score for *RES2* is related to the run that obtained the best results (run 1).

The baseline has been computed by running a simple zero-shot prompt on Qwen2.5-7B-Instruct model⁴; the prompt used is reported in appendix A.

kenji-endo obtained the highest score on subtask 1, while *RES2* had the best performances on subtask 2. Subtask 3 has been addressed by *RES2* only.

Team	F1 subtask 1	F1 subtask 2	F1 subtask 3
kenji-endo	0.9496	0.2226	-
RES2	0.9089	0.8762	0.6369
Baseline	0.4602	0.3352	0.2942

Table 4

Macro F1 score obtained by the participants on the three subtasks.

4.2.1. Subtask 1

Both the participants obtained high scores on the implicit detection task. Figure 1 reports the confusion matrices related to *kenji-endo* and *RES2*. Only one run of *RES2* system is reported, given that the results of the two runs are identical in this subtask.

In general, *kenji-endo* ($F1 = 0.9496$) perform better than *RES2*, with a lower number of wrong assignment in both the classes (0 and 1).

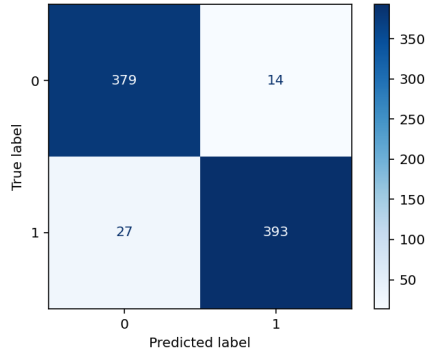
4.2.2. Subtask 2

In the implicit classification task, *RES2* obtained the highest score ($F1 = 0.8762$), and a low number of errors in both the classes (IMP and PPP). Conversely, the system by *kenji-endo* classified all the occurrences as IMP, obtaining a poor result, below the baseline. Figure 2 reports the confusion matrices. Again, only one run of *RES2* system is reported, given that the results of the two runs are identical in this subtask.

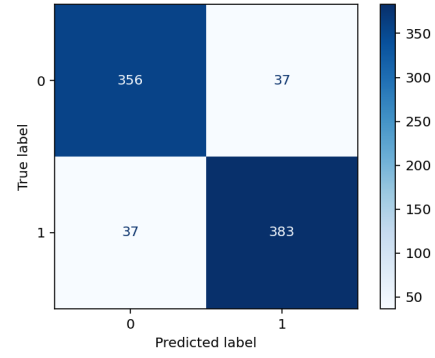
4.2.3. Subtask 3

In the implicature classification task, *RES2* was the unique participant, obtaining $F1$ score = 0.6369. Figure 3 reports the confusion matrices of the two runs, that obtained slightly different scores on this

⁴<https://huggingface.co/Qwen/Qwen2.5-7B-Instruct>

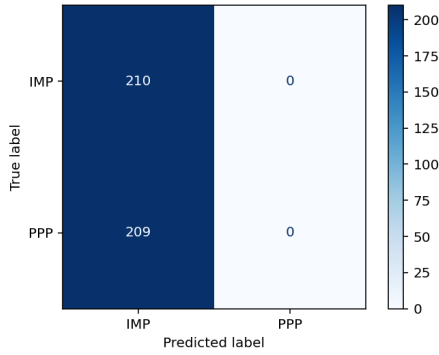


Team *kenji-endo*

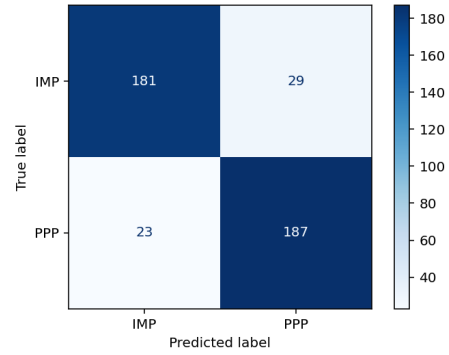


Team *RES2*

Figure 1: Confusion matrices for subtask 1



Team *kenji-endo*

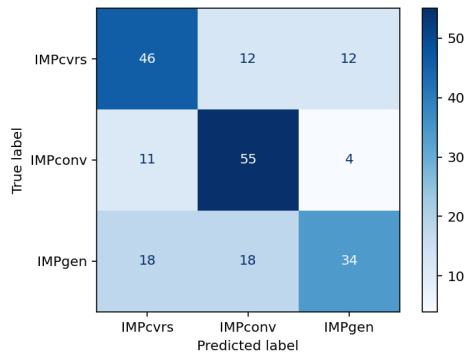


Team *RES2*

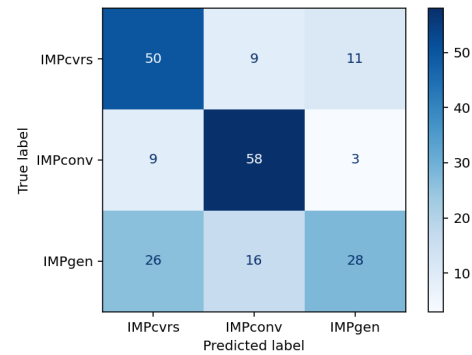
Figure 2: Confusion matrices for subtask 2

subtask. The first run had a higher F1 score.

Interestingly, the error distribution among the three classes (IMPcvrs, IMPconv, and IMPgen) is quite unbalanced: conventional implicatures are well identified (F1 for IMPconv = 0.71), while *RES2* system performed worse on conversational implicatures (F1 for IMPcvrs = 0.63; F1 for IMPgen = 0.57).



Team *RES2* - run 1



Team *RES2* - run 2

Figure 3: Confusion matrices for subtask 3

5. Discussion

Implicit content detection and classification are non-trivial tasks, that can not be easily addressed by an LLM with a simple prompt. This is confirmed by the poor results obtained with the baseline, reporting a low F1 score across all subtasks. The three proposed subtasks have an increasing level of specificity and complexity: the more general implicit detection task has been solved by both the participants, while the discrimination between implicature and presupposition proved to be more challenging, and the implicature type identification was even harder. In addition to this, there is a different number of items in the training sets: the dataset for subtask 2 is about half the size of that for subtask 1, and the dataset for subtask 3 is about half of that for subtask 2. These differences may have impacted the effectiveness of the training, making subtask 3 harder to address.

To gain further insight into the results, we conducted a qualitative error analysis on the outputs of both participants. Our aim was to identify systematic patterns in the mistakes across the three subtasks and to determine which types of implicit content posed the greatest challenges.

For **subtask 1**, we examined errors from both participants. Across the dataset, 12 instances were consistently misclassified by both teams. Among these, one was a non-implicit expression, while the remaining 11 were implicit, including 9 presuppositions and 2 implicatures. At the participant-level, *kenji-endo*'s model produced 12 errors involving presuppositions and 15 involving implicatures, 14 of which were conversational, indicating a strong reliance on contextual interpretation. Errors produced by *RES2* were mostly related to presuppositions (27 out of 37), which in the corpus are largely sub-categorized. 12 of these errors involved definite descriptions, which introduce discourse entities as identifiable by the hearer through lexical-syntactic indexes such as personal pronouns and demonstratives. On the other hand, the majority of implicature misclassification concerned conversational ones. Overall, errors in subtask 1 were primarily driven by difficulties in identifying implicit content, particularly presuppositions.

The *non-bona fide* nature of the implicit content targeted by our task - closely related to the communicative setting in which the discourse is produced - made presuppositions particularly difficult to detect, even when relying on lexical-syntactic cues. In line with this observation, context-dependent implicatures remain a major challenge, likely because they require pragmatic inference, and discourse-level knowledge. By contrast, conventional implicatures, being lexically or grammatically encoded, were more easily classified by the system.

For **subtask 2**, the analysis focused on *RES2*, since *kenji-endo*'s system exhibited considerably lower performance and classified all the occurrences as IMP. Only two of *RES2*'s errors overlapped with those observed in subtask 1, suggesting a limited correlation between detection and classification errors. *RES2* misclassified 54 instances: 23 presuppositions (PPP) and 31 implicatures (IMP). Among presuppositions, no clear preference for specific sub-categories emerged. Conversely, among implicatures, 29 were particularized conversational and only 2 were conventional. This distribution further confirms that conversational implicatures represent the most challenging category.

For **subtask 3**, the analysis again focused on the best-performing run of *RES2*. Errors followed a pattern similar to that observed in subtask 2, with conversational implicatures proving more difficult to classify than conventional ones. Notably, 60% of generalized implicatures were misclassified, highlighting the complexity of this category and its resistance to reliable automatic classification.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

- [1] H. P. Grice, Logic and conversation, in: P. Cole, J. L. Morgan (Eds.), *Syntax and Semantics 3: Speech Acts*, Academic Press, 1975, pp. 41–58.
- [2] R. Stalnaker, Assertion, *Syntax and Semantics 9* (1978) 315–332.
- [3] P. Chilton, Manipulation, Persuasion and Propaganda, John Benjamins, 2005.
- [4] E. L. Vallauri, Implicit meaning and persuasion, *Journal of Pragmatics* (2019).
- [5] F. Cominetti, L. Gregori, E. Lombardi Vallauri, A. Panunzi, et al., Impaqts: un corpus di discorsi politici italiani annotato per gli impliciti linguistici, in: *Corpora e Studi linguistici. Atti del LIV Congresso della Società di Linguistica Italiana* (Online, 8–10 settembre 2021), a cura di Emanuela Cresti e Massimo Moneglia. Milano, Officinaventuno, 2022, pp. 151–164.
- [6] F. Cominetti, L. Gregori, E. L. Vallauri, A. Panunzi, Impaqts: a multimodal corpus of parliamentary and other political speeches in Italy (1946–2023), annotated with implicit strategies, in: *Proceedings of the IV Workshop on Creating, Analysing, and Increasing Accessibility of Parliamentary Corpora (ParlaCLARIN)@ LREC-COLING 2024*, 2024, pp. 101–109.
- [7] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for Italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [8] S. Spezia, S. Ferrara, R. Dioguardi, E. Davi, I. Siragusa, R. Pirrone, RES2 at IMPOLS 2026: Implicit content classification via span-awareness, hierarchical multi-task learning and synthetic data, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [9] C. J. Scozzaro, M. Rinaldi, G. Mittone, M. A. Stranisci, Kenji-Endo at IMPOLS 2026: A BabyLM, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.

A. Prompts used for baseline computation

A.1. Prompt for subtask 1

Original Italian prompt used for the baseline

Considera il seguente estratto di discorso politico:

[text]

Il testo contenuto tra <s> e </s> contiene un implicito non bona fide vero? Rispondi solo sì o no, senza aggiungere altro testo.

English translation

Consider the following excerpt from a political speech:

[text]

Does the text contained between <s> and </s> contain an implicit meaning that is not bona fide true? Answer only yes or no, without adding any other text.

A.2. Prompt for subtask 2

Original Italian prompt used for the baseline

Considera il seguente estratto di discorso politico:

[text]

Il testo contenuto tra <s> e </s> contiene un implicito non bona fide vero. Di che tipo è l'implicito? Rispondi PPP se ritieni che sia una presupposizione, IMP se ritieni che sia un'implicatura. Non aggiungere altro testo.

English translation

Consider the following excerpt from a political speech:

[text]

The text contained between <s> and </s> contains an implicit meaning that is not bona fide true. What type of implicit meaning is it? Answer PPP if you believe it is a presupposition, IMP if you believe it is an implicature. Do not add any other text.

A.3. Prompt for subtask 3

Original Italian prompt used for the baseline

Considera il seguente estratto di discorso politico:

[text]

Il testo contenuto tra <s> e </s> contiene un'implicatura. Di che tipo è l'implicatura? Rispondi CVRS se ritieni che sia un'implicatura conversazionale particolarizzata, GEN se ritieni che sia un'implicatura conversazionale generalizzata, CONV se ritieni che sia un'implicatura convenzionale. Non aggiungere altro testo.

English translation

Consider the following excerpt from a political speech:

[text]

The text contained between <s> and </s> contains an implicature. What type of implicature is it? Answer CVRS if you believe it is a particularized conversational implicature, GEN if you believe it is a generalized conversational implicature, CONV if you believe it is a conventional implicature. Do not add any other text.

B. Implicit examples

Below, one example is provided for each category to be identified in the three subtasks, including the original Italian transcription and its English translation. The sentence to be annotated is shown **in bold**. Where implicit content is involved, a brief explicit explanation is provided in italics for a better task understanding; however, such explanations are not provided in the IMPOLS dataset.

B.1. Examples for subtask 1

Speech without questionable implicit contents

Original Italian:

Resterò vicino al cemento e agli sforzi dell' Italia e degli italiani, con infinita gratitudine per quel che ho ricevuto in questi quasi 9 anni non soltanto di riconoscimenti legati al mio ruolo, non soltanto di straordinarie occasioni di allargamento delle mie esperienze, anche internazionali, **ma per quel che ho ricevuto soprattutto di espressioni di generosa fiducia e costante sostegno, di personale affetto, direi, da parte di tantissimi italiani che ho incontrato o comunque sentito vicini.**

[Giorgio Napolitano, 2014]

English translation:

I will remain close to the feat and efforts of Italy and of the Italian people, with endless gratitude for what I have received over these almost nine years not only in terms of recognition related to my role, not only in terms of extraordinary opportunities to broaden my experience, including at the international level, **but above all for what I have received in the form of expressions of generous trust and constant support, and, I would say, of personal affection, from the many Italians I have met or have nevertheless felt close to.**

Speech with questionable implicit content

Original Italian:

Recuperiamo, recuperiamo ciascuna componente con la sua identità, il meglio delle nostre tradizioni, di quella socialista e comunista, di quella cattolico-popolare, di quella laica e repubblicana. **I nostri partiti sono altra cosa rispetto ai loro antecedenti storici, le forze schierate da questa parte rappresentano la rottura più conseguente col vecchio sistema politico e di potere.**

Implica che le attuali forze di centro-sinistra siano migliori rispetto ai propri antecedenti.

[Giorgio Napolitano, 1996]

English translation:

Let us recover each component with its own identity, the best of our traditions, the socialist and communist tradition, the Catholic-popular one, and the secular and republican one. **Our parties are something different from their historical predecessors, the forces aligned on this side represent the most coherent break with the old political and power system.**

It implies that the current centre-left forces are better than their predecessors.

B.2. Examples for subtask 2

Implicature

Original Italian:

Questa non è una crisi qualsiasi, è la fine del grande imbroglio italiano che dura da ben 15 anni. È stato un imbroglio il golpe mediatico-giudiziario che ha portato al governo del Paese con la forza e la violenza **chi, dopo il crollo del muro di Berlino, è stato sconfitto dalla storia.**

Implica che la sinistra sia stata sconfitta dalla storia.

[Stefania Craxi, 2007]

English translation:

This is not just any crisis, it is the end of the great Italian fraud that has lasted for a full fifteen years. It was a fraud the media and judicial coup that brought to power, through force and violence, **those who, after the fall of the Berlin Wall, were defeated by history.**

It implies that the left was defeated by history.

Presupposition

Original Italian:

Queste sono le prime elezioni veramente libere di tutto il Dopoguerra, perché per la prima volta si vota liberi dalla paura del Comunismo che aveva condizionato tanta parte della vita politica italiana dal '45 in poi, e che aveva condizionato soprattutto le scelte del mondo cattolico. Oggi si può votare dunque secondo coscienza, **non più turandosi il naso come tante volte aveva dovuto fare tanta parte dell' elettorato**, anche di sentimenti nazionali.

Presuppone che tanta parte dell'elettorato abbia dovuto votare turandosi il naso.

[Pino Rauti, 1990]

English translation:

These are the first truly free elections of the entire post-war period, because for the first time people are voting free from the fear of Communism, which had shaped a large part of Italian political life since '45 and had especially influenced the choices of the Catholic world. Today, therefore, one can vote according to conscience, **no longer holding one's nose as a large part of the electorate had so often been forced to do**, including voters with strong national sentiments.

It presupposes that a large part of the electorate had been forced to vote while holding its nose.

B.3. Examples for subtask 3

Conventional implicature

Original Italian:

I partiti presentavano, con un programma, quello che volevano fare prima delle elezioni; poi, una volta al governo, facevano tutt'altro. La Legge Fornero non era in nessun programma elettorale, nessun italiano l'ha mai votata. **Il Jobs Act non era in nessun programma, eppure l'hanno fatto.**

Implica che tutti i governi dovrebbero rispettare i programmi elettorali, che non dovrebbero fare niente che non sia scritto nel programma elettorale.

[Luigi Di Maio, 2018]

English translation:

Before elections, parties would present a programme outlining what they intended to do; then, once in government, they would do something entirely different. The Fornero Law was not included in any electoral programme, no Italian ever voted for it. **The Jobs Act was not included in any programme, yet they implemented it.**

It implies that all governments should respect their electoral programmes and should not implement measures that are not included in them.

Particularized conversational implicature

Original Italian:

Chi non la voleva capire non l' ha capita, eppure il significato è chiaro, non è la politica sottobanco che voglio fare, non è la politica sottobanco che appassiona le persone come voi, come noi. E nemmeno i patti dell'amministrazione con i poteri forti che innalzano quartieri in mezzo al nulla **invece di realizzare servizi e trasporti che rendono la vita più semplice ai nostri cittadini.**

Implica che l'amministrazione Alemanno non abbia realizzato alcun servizio o trasporto che rendesse la vita più semplice ai cittadini.

[Ignazio Marino, 2013]

English translation:

Those who did not want to understand it did not understand it, yet the meaning is clear. This is not the kind of backroom politics I want to pursue, nor the kind that engages people like you and like us. Nor are they the deals between administrations and vested interests that raise neighbourhoods in the middle of nowhere **instead of providing services and transport that make life easier for our citizens.**

It implies that the Alemanno administration did not provide any services or transport that made life easier for citizens.

Generalized conversational implicature

Original Italian:

E allora, alla scelta di dire "no" all'eutanasia, si accompagna la scelta di dire "sì" alla medicina palliativa, "sì" ad una compagnia che aiuti ad affrontare in modo umano la morte. Ultima cosa, il paziente ha diritto alla protezione contro il dolore. **Molti credono di essere a favore dell'eutanasia** perché non sopportano l'idea di morire in mezzo a terribili sofferenze, o di veder morire tra terribili sofferenze una persona amata.

Implica che molti di quelli che sostengono di essere a favore dell'eutanasia non lo siano veramente.

[Rocco Buttiglione, 2011]

English translation:

And so, alongside the choice to say "no" to euthanasia comes the choice to say "yes" to palliative care, "yes" to companionship that helps people face death in a humane way. One final point, patients have the right to protection from pain. **Many believe they are in favour of euthanasia** because they cannot bear the idea of dying in terrible suffering, or of seeing a loved one die in terrible suffering.

It implies that many of those who claim to be in favour of euthanasia are not truly in favour of it.

RES2 at IMPOLS 2026: Implicit Content Classification via Span-Awareness, Hierarchical Multi-Task Learning and Synthetic Data

Salvatore Spezia^{1,†}, Salvatore Ferrara^{1,†}, Roberto Dioguardi^{1,†}, Ernesto Davì^{1,†}, Irene Siragusa^{1,*} and Roberto Pirrone¹

¹Department of Engineering, University of Palermo, Viale delle Scienze, Edificio 6, Palermo, 90128, Sicily, Italy

Abstract

In this report, we present the proposed approach of the RES2 team for the multimodal IMPOLS challenge at EVALITA 2026 campaign. We used a text-only approach based on UmBERTo Language Model, enhanced with a Span-Aware Architecture, which concatenates both global and local dense representations. Due to the scarcity of the data set, we implemented a Span-Preserving Data Augmentation with Gemini 3.0 and a Hierarchical Multi-Task Learning Strategy. The proposed system won in subtasks 2 and 3 and reached the 2nd place in subtask 1. In addition, we document the exclusion of the audio data, which resulted ineffective due to misalignment issues, speaker bias, and insufficient discriminative power.

Keywords

Span-Aware Transformers, Hierarchical Multi-Task Learning, Implicit Content Detection, Political Speech, Language Models

1. Introduction

Political discourse is a highly strategic field of human communication in which persuasion often relies on implicit rather than explicit meaning. The use of linguistic strategies such as presuppositions and implicatures enables speakers to convey questionable or manipulative content, defined as *non-bona fide* by bypassing the critical evaluation of the listeners and reducing their own assertive responsibility [1].

In the context of the EVALITA 2026 campaign [2], the Implicit Content in Political Speech Task (IMPOLS) [1] focuses on the development of automatic recognition systems for implicit content in Italian political speeches for both textual and audio data. The complexity of the task lies in the subtle nature of these mechanisms, as can be seen in the data set of the challenge extracted from IMPAQTS corpus [3].

Implicit content is not determined only by the words used, but by the interaction between a specific *local linguistic trigger* and the surrounding pragmatic context. The challenge is divided into three different subtasks, all evaluated in terms of macro F1-score, for binary detection of implicit content (subtask 1), binary classification tasks for implicatures and presuppositions (subtask 2) and implicature multi-class classification (subtask 3). For all three subtasks, the target segment which must be classified is annotated.

Recent progresses in Natural Language Processing (NLP) field are mainly led by generative Large Language Models (LLMs) but they struggle to capture fine-grained pragmatic nuances in zero-shot, few-shot, and Chain-of-Thought (CoT) settings, particularly in languages other than English [4]. Furthermore, the availability of annotated data for such specific phenomena is limited.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ salvatore.spezia@community.unipa.it (S. Spezia); salvatore.ferrara05@community.unipa.it (S. Ferrara); roberto.dioguardi@community.unipa.it (R. Dioguardi); ernesto.davi@community.unipa.it (E. Davì); irene.siragusa02@unipa.it (I. Siragusa); roberto.pirrone@unipa.it (R. Pirrone)

ORCID 0009-0005-8434-8729 (I. Siragusa); 0000-0001-9453-510X (R. Pirrone)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

In this paper we present the system developed by the RES2 team for the three proposed subtasks in the IMPOLS challenge at EVALITA 2026, which employs a text-only strategy involving *ad hoc* fine-tuned version on UmBERTo [5] encoder-only Language Models (LMs). The core idea was to focus on both the global context and the local trigger to detect and classify the pragmatic ambiguities in the given speech.

The main contributions of this work can be summarized as follow:

1. A **Span-Aware Architecture** which concatenates the global representation of the input (Global Context Vector) with the target utterance representation (Local Target Vector), following a span-aware strategy;
2. A **Span-Preserving Data Augmentation Strategy** to mitigate the scarcity of samples for subtasks 2 and 3 with Gemini 3.0 [6], which paraphrases the only context while preserving the integrity of linguistic triggers;
3. A **Hierarchical Multi-Task Learning strategy** involving the hierarchical structure of the three tasks to lead the training of complex features through a weighted loss;
4. An analysis over the audio data, which highlights how the speaker bias tends to override useful prosodic signals, thereby justifying our final selection of a uni-modal approach.

The proposed system reached the 1st place in both subtasks 2 and 3 and the 2nd place in subtask 1, assessing the validity of the proposed approach against a general-purpose LLM.

This report is arranged as follows: Section 2 describes the proposed system along with data augmentation process, while details regarding the experimental setup are in Section 3. Obtained results and related discussion are reported in Section 4 and 5. Concluding remarks are in Section 6.

2. Description of the system

A unique transformers-based encoder-only model [7] has been developed for all the three subtasks. We opt for UmBERTo [5] as the LM backbone of the proposed system, which have been adapted to the target political speech domain, following a Layer-Wise fine-tuning strategy, differentiated for each subtask.

Given the pragmatic nature of the challenge, where implicit content is often activated by specific linguistic triggers within a broader context, we hypothesized that the standard [CLS] token alone would be insufficient to capture the necessary linguistic nuances. Thus, we propose a Span-Aware approach that combines the global semantic representation of the utterance with a focal representation of the target span. Furthermore, to mitigate data scarcity in subtask 3, we introduced an LLM-based Span-Preserving Data Augmentation strategy and a Hierarchical Multi-Task Learning framework.

The proposed Span-Aware Architecture (Figure 1) concatenates global and local feature, thus enhancing the capabilities of the overall system, which can be aware of both the target utterance and its context. In the proposed methodology, instead of relying exclusively on the features associated with the [CLS] token, we make the model explicitly Span-Aware. Specifically, the model is designed to be sensitive to a text portion delimited and annotated as the one which carries an implicit content, following the principle that the direct representation of spans enhances semantic modeling [8].

Formally, given the input sequence $X = \{x_1, \dots, x_n\}$, we add two special delimiter tokens <TGT_START> e <TGT_END> to mark the span target in the given textual input, following strategies based on explicit markers to highlight mentions or spans within transformer-based architectures [9]. Let $H \in \mathbb{R}^{n \times d}$ output of the last hidden state of the encoder, where d is the embedding size, which is equal to 768 for UmBERTo LM.

Then, we extract the two global and local embedding representations:

- **Global Context Vector $\mathbf{h}_{cls}$** corresponds to the embedding of the [CLS] token (H_0);
- **Local Target Vector $\mathbf{h}_{span}$** obtained through a mean pooling operation over the embeddings of the token enclosed between the previously defined delimiter tokens, as in [9];

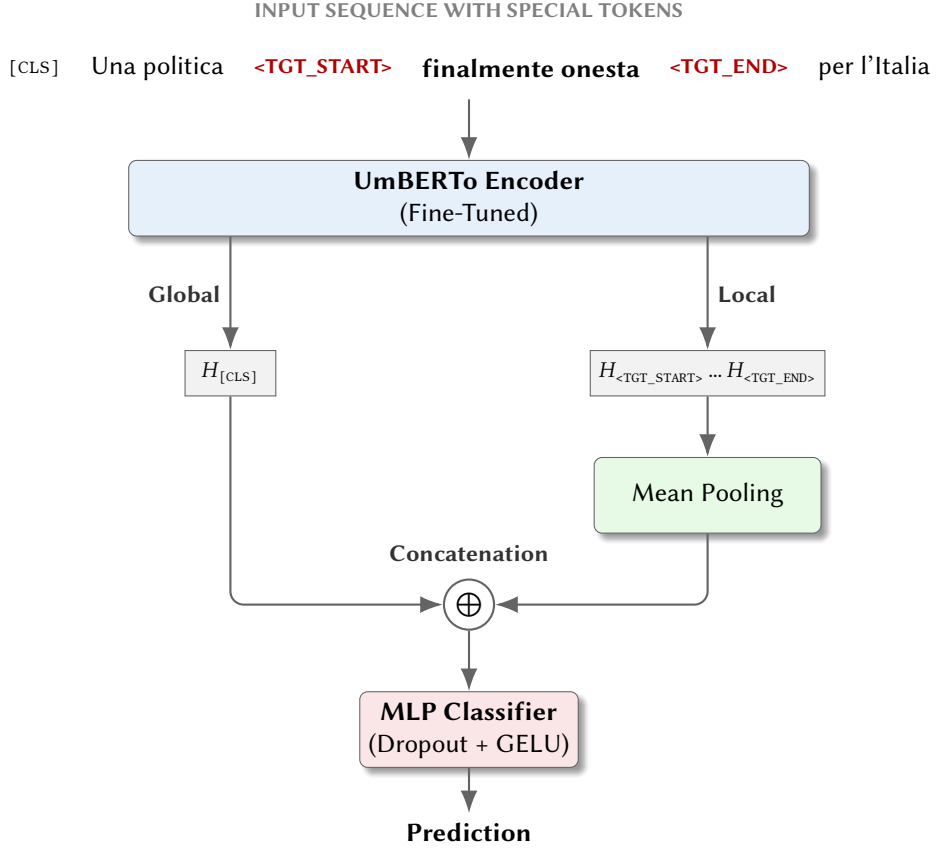


Figure 1: Architecture of the proposed Span-Aware model. The global $[CLS]$ embedding $\mathbf{h}_{cls} \in \mathbb{R}^d$ is concatenated with the mean-pooled target-span representation $\mathbf{h}_{span} \in \mathbb{R}^d$ (tokens between $\langle TGT_START \rangle$ and $\langle TGT_END \rangle$), producing $\mathbf{h}_{final} = [\mathbf{h}_{cls}; \mathbf{h}_{span}] \in \mathbb{R}^{2d}$, which is fed to an MLP classifier.

Thanks to a concatenation operation, the final representation is obtained as $\mathbf{h}_{final} = \mathbf{h}_{cls} \oplus \mathbf{h}_{span}$, which serves as an input to a Multi-Layer Perceptron (MLP) classifier which uses a dropout and a GELU [10] activation function to calculate the final prediction. On doing this, the model can use both the linguistic trigger and the overall context to calculate the final prediction.

In the context of the IMPOLS shared-task, target trigger span boundaries are provided by the organizers as part of the annotated input. Accordingly, at inference time our approach conditions on these gold spans by inserting the $\langle TGT_START \rangle$ and $\langle TGT_END \rangle$ markers, and we do not address the upstream problem of automatically localizing trigger spans. We are aware that this could be a limitation in real-case scenario of the proposed system. In this case, an additional target detection method should be included in the pipeline as pre-processing step on the textual input, before it is injected in the proposed system.

In the following subsections, we will deeply focus on the data augmentation phase (Sections 2.1) and the hierarchical multi-task learning strategy to address subtask 3 (Sections 2.2).

2.1. Span-Preserving Data Augmentation

Although the IMPOLS data set has balanced classes, the hierarchical nature of the challenge leads to a severe contraction in data volume as one progresses toward the more fine-grained subtasks. Starting from 1,824 samples in subtask 1, availability is halved to 912 for subtask 2 and is further reduced to only 456 examples for subtask 3.

This scarcity represents a critical bottleneck that exposes models to a high risk of overfitting. To mitigate this problem in the context of subtasks 2 and 3, we implemented a synthetic data augmentation pipeline using Gemini 3.0 [6]. On doing this, we incremented the cardinality of the training set and the

generalization capabilities of the model, making it robust to lexical variations.

In contrast to standard paraphrasing, we adopted a Span-Preserving approach, in which we ensured that the generative models do not change the trigger, with the following prompt:

Riscrivi il contesto e la frase circostante per variare lessico e sintassi, ma mantieni inalterato lo span target originale che attiva l'implicatura

With this constraint we prevent the generative model from modifying the trigger, e.g. substitute *purtroppo* with *sfortunatamente* may change the implicit nature of the given sentence or its class [11]. Two variations have been created for each sample in training sets for subtasks 2 and 3, thus tripling the size of the available training sample.

An ablation study (Section 4.1) deeply explores the impact of the proposed data augmentation strategy and confirms its positive impact of performances over subtasks 2 and 3, with a particularly large improvement on subtask 3.

2.2. Hierarchical Multi-Task Learning

Limited to subtask 3, which is a multi-class classification task of particularized and generalized conversational, or conventional implicatures, we involved the inherent hierarchical and coarse-to-fine structure of the three subtasks in the IMPOLS challenge. Specifically, subtask 3 presupposes the distinction between implicature and presupposition (subtask 2), which in turn depends on identifying the presence of implicit content (subtask 1), consistent with a hierarchical classification framework [12].

Therefore, we adopted a Multi-Task Learning (MTL) approach using an hard parameter sharing strategy. We trained a single model with three independent classification heads (e.g. three different MLP Classifiers), jointly trained from scratch and sharing the same Span-Aware encoder. On doing this, an inductive transfer strategy is implemented, in which the supervision of correlated tasks guide the learning of more robust and generalizable dense representations [13].

Since the main focus was towards increasing the performances over subtask 3, subtasks 1 and 2 are used as auxiliary tasks, in an auxiliary-task learning approach [14]. In this case, the coarse grained tasks, semantic correlated to the target one, serve as stabilization factor during the training phase, and as improving factor for the overall performance. Hence, this strategy acts as a scaffolding structure for the effective learning of feature for the complex fine-grained classification in subtask 3.

Consequently, the loss function $\mathcal{L}_{tot}$ is defined as the weighted sum of the Cross-Entropy Loss of each subtask:

$$\mathcal{L}_{tot} = \lambda_1 \mathcal{L}_{task1} + \lambda_2 \mathcal{L}_{task2} + \lambda_3 \mathcal{L}_{task3}$$

We defined the following constraint $\lambda_3 \gg \lambda_1, \lambda_2$, to highlight the focus on subtask 3. Based on preliminary experiments on the validation set, we set the weights to $\lambda_1 = 1.5$, $\lambda_2 = 0.5$ e $\lambda_3 = 5.0$.

3. Experimental Setup

The data set of the IMPOLS challenge is derived from IMPAQTS [3], a multimodal corpus collecting Italian political speeches. Following a hierarchical division, the three perfectly balanced data sets for each task has been built.

We used UmBERTo-commoncrawl-cased-v1 [5] as the LM backbone of the proposed system. UmBERTo is a LM based on RoBERTa [15] architecture and trained on an Italian web crawling corpus. We decided to use UmBERTo since (i) it employs SentencePiece [16] as tokenizer, which better manages the morphological complexity of the Italian language compared with standard WordPiece, and (ii) applies the Whole Word Masking during its pre-training phase, which preserves the semantic integrity of the concepts.

Since the semantic complexity of the task and the small training set, we opted for a Stratified K-Fold Cross-Validation, with K=5, instead of a fixed custom train-validation split. Within this strategy, the ensemble obtained model resulted to be much more stable and robust.

We adopted an hyperparameter optimization strategy, differentiated for each task: a classic grid search for subtask 1, while a bayesian optimization with Optuna [17] combined with a manual refinement for subtasks 2 and 3. At training time, our scope was towards the maximization of the macro F1-score. The final hyperparameters used for each subtask are summarized in Table 1.

Table 1

Overview of the used hyperparameters. Values marked with (*) were determined experimentally without automated search.

Hyperparameter	Subtask 1	Subtask 2	Subtask 3
Optimizer	AdamW*	AdamW*	AdamW*
Learning Rate	2×10^{-5}	1.31×10^{-4}	4.5×10^{-5}
Batch Size	16	8*	8*
Accumulation Step	-	2	2
Max Epochs	12*	15*	20*
Unfrozen Layers (Last N)	4	6	8
Dropout	0.20	0.15	0.25
Weight Decay	0.02	0.05	0.20
Label Smoothing	-	0.10	0.15
MTL Loss Weights ($\lambda_1, \lambda_2, \lambda_3$)	-	-	1.5, 0.5, 5.0
Hidden MLP Size	768*	128*	256

To adapt the LM to the target political speech domain, we used a Layer-Wise fine-tuning strategy, differentiated for each subtask. In particular, we trained the last 4 transformers layers for subtask 1, the last 6 layers for subtask 2 and the last 8 for subtask 3. The number of trained layers has been empirically found as the optimal balance between the number of training samples for each subtask and their complexity.

3.1. Training Details

Developed architectures were trained using Google Colab instances equipped with one GPU Tesla T4 and a local machine with NVIDIA GeForce RTX 5070 GPU. Models were implemented in PyTorch [18] incorporating stabilization techniques such as Mixed Precision (FP16), Early Stopping (with patience set to 3–5 epochs), and a linear scheduler with warm-up.

Starting from the developed and trained models, following a Stratified 5-Fold Cross-Validation strategy, the final model has been obtained as an aggregation of the 5 folds through Logit Averaging. For binary classification tasks, the final prediction is obtained by applying a sigmoid function to the mean of the logits and a 0.5 threshold:

$$\hat{y}_{bin} = \mathbb{I} \left(\sigma \left(\frac{1}{K} \sum_{k=1}^K \mathbf{z}_k \right) > 0.5 \right)$$

A softmax function is applied for the multi-class classification task to the mean of the logits, and the predicted label is the one with the highest probability.

$$\hat{y}_{multi} = \operatorname{argmax} \left(\operatorname{softmax} \left(\frac{1}{K} \sum_{k=1}^K \mathbf{z}_k \right) \right)$$

where $K = 5$ and $\mathbf{z}_k$ is the logits vector of the k -th model. On doing this, we highly reduced the prediction variance among different folds.

4. Results

In Table 2 are reported the official result over the test set of the IMPOLS challenge [1].

Table 2

Official result over the test set (macro F1-score).

System	Subtask 1	Subtask 2	Subtask 3
baseline	0.4602	0.3352	0.2942
kenji-endo	0.9496	0.2226	-
RES2 (ours)	0.9089	0.8762	0.6369

In [1] a simple zero-shot prompt on Qwen2.5-7B-Instruct model [19] has been proposed as baseline for all three subtasks. The proposed RES2 system, overcome the baseline for all subtasks, reaching the 2nd place in the subtask 1, with a macro F1-score of 0.9089, resulting an improvement of 98% to the baseline model. As for subtasks 2 and 3, RES2 model reached the 1st place in the rank with almost tripling and doubling the scores obtained by the baseline, with 0.8762 and 0.6369 as macro F1-score. Obtained results assess the overall validity of the proposed architecture based on encoder-only models, along with the proposed MTL and data augmentation strategies.

During the development phase, we evaluated the performance of each fold over the corresponding validation set, which results are reported in Table 3. There can be highlighted the robustness of the proposed approach due to a small variance among different models, which reached its peak of $\sigma \approx 0.03$ for subtask 3, the most complex one.

Table 3

Average performance over the validation sets of the Stratified 5-Fold Cross-Validation

Macro F1-score (mean $\pm$ std)	
Subtask 1	0.9803 $\pm$ 0.0053
Subtask 2	0.8252 $\pm$ 0.0257
Subtask 3	0.7165 $\pm$ 0.0261

4.1. Ablation Study: Impact of Data Augmentation

To evaluate the impact of the proposed span-preserving data augmentation in subtasks 2 and 3, we trained the model with the only IMPOLS data and compared the obtained results with the performances of the model trained with the augmented data, while keeping unchanged the remaining hyperparameters.

Table 4

Ablation study on span-preserving data augmentation in subtasks 2 and 3 (macro F1-score), reporting both validation (Average Stratified 5-Fold CV) and official test results.

Strategy	Validation		Test	
	Subtask 2	Subtask 3	Subtask 2	Subtask 3
w/o augmentation	0.8015	0.5892	0.8666	0.4106
w/ augmentation	0.8252	0.7165	0.8762	0.6369
Δ (w/ - w/o)	+0.0237	+0.1273	+0.0096	+0.2263

Results reported in Table 4, show consistent improvements, with a particularly large gain on subtask 3, where data scarcity is most severe, thus assessing the positive impact of the proposed data augmentation strategy.

5. Discussion

In this section we will explore the impact of diverse encoder-only models as backbone of the developed Span-Aware architecture (Section 5.1) and its related architectural choices (Section 5.2), and our analysis of the audio data in the IMPOLS data set and our test within audio-only (Section 5.3) and multimodal models (Section 5.4).

5.1. Analysis over the LMs backbone

The proposed model is the best one developed during the training phase, in which we tested other LMs backbones with the proposed Span-Aware architecture. Specifically, we analyzed BERT-it cased (dbmdz/bert-base-italian-cased) [20], mDeBERTa-v3 (microsoft/mdeberta-v3-base) [21], XLM-RoBERTa (FacebookAI/xlm-roberta-base) [22] and GilBERTo (idb-ita/gilberto-uncased-from-camembert) [23], in addition to UmBERTo. With reference to the subtask 2, in Table 5 are reported the obtained results with the diverse LMs backbone under analysis.

Table 5

Comparisons of the performances of the LMs backbones over subtask 2 over the validation sets of the Stratified 5-Fold Cross-Validation. Bold values marks best obtained results.

Backbone Model	Macro F1-score (mean $\pm$ std)
BERT-it	0.7850 $\pm$ 0.0226
mDeBERTa-v3	0.8029 $\pm$ 0.0135
XLM-RoBERTa	0.8070 $\pm$ 0.0304
GilBERTo	0.8134 $\pm$ 0.0155
UmBERTo	0.8252 $\pm$ 0.0257

The analysis and the obtained results show that UmBERTo lightly overpass both multilingual models (mDeBERTa-v3 and XLM-RoBERTa) and Italian-only models (GilBERTo and BERT-it), thus it was selected our LM backbone.

5.2. Analysis over the Span-Aware architecture

The final developed method involved the simple concatenation $\mathbf{h}_{cls} \oplus \mathbf{h}_{span}$, we also explored other methodologies involving attention pooling and a self-attention or cross-attention layer to merge the two vector representation. This resulted in lower performances in the training set (macro F1-score of 0.73 for subtask 2) and an unnecessary computational complexity.

Another strategy was to consider $\mathbf{h}_{cls}$ and $\mathbf{h}_{span}$ as distinct feature maps which can be combined with a Convolutional layer (CNN). This idea was promising and competitive in subtask 1, reaching a score in the training set was equal to 0.9562, but still lower compared with the simple concatenation (0.9803).

In conclusion, comparing the results obtained during the development phase (Table 3) with the official competition results (Table 2), it is possible notice that the proposed system does not suffer from overfitting, despite the highest results with the training set. We assume that its generalization capabilities reside in both the system architecture and training strategy implemented, comprehensive of the proposed data augmentation strategy. The simple combination of UmBERTo as the LM backbone and a basic feature concatenation, resulted to be more effective than other methodologies, which may require additional computational resources and an increasing number of parameters.

While conditioning on gold trigger spans is reasonable in the IMPOLS shared-task scenario, it limits direct applicability in real-world settings where such annotations are unavailable. In practical deployments, the proposed classifier would need to be preceded or jointly coupled with an automatic trigger-span detection component. Assessing robustness under automatically predicted spans is left for future work.

5.3. Analysis of the multimodal data set

The multimodal nature of the IMPAQTS data set [3] offered, at least theoretically, the opportunity to leverage paralinguistic cues, such as prosody, intonation and pauses, to disambiguate implicit content. However, before finalizing the text-only architecture described in Section 2, we conducted an exploratory analysis and a set of audio-only probes on the acoustic modality.

A preliminary inspection revealed a non-negligible misalignment between the audio tracks and the provided transcriptions. Through a semi-automated verification procedure based on segment duration, we excluded 110 samples (approximately 6% of the original data set) exhibiting a complete mismatch. This reduction introduces an initial limitation for the audio modality compared to text, which is instead fully preserved.

To quantify the contribution of audio, we extracted 345 acoustic features for each trigger-associated segment, combining prosodic descriptors (e.g., F0, jitter, shimmer) via Praat [24], spectral features (e.g., MFCC and spectral centroid) via Librosa [25], and the standard eGeMAPSv02 set [26] using OpenSMILE [27]. Moreover, to mitigate inter-speaker variability, in addition to the speaker-level normalization, we adopted an intra-speaker local baseline comparison. For each trigger segment we consider the same-duration baseline window within the same recording, and we define delta features Δ as the difference between segment and baseline features. This representation aims to subtract speaker-specific prosodic baselines and capture local acoustic variations around the trigger.

To assess speaker bias and estimate the actual discriminative power of acoustic features with respect to the task labels, we trained a Random Forest classifier under two 5-cross-validation schemes. Specifically we considered (i) a stratified K-Fold (stratified CV), consistent with the text-only development setting, and (ii) Speaker-Aware K-Fold (speaker CV), ensuring that all segments from the same speaker fall within the same fold. Table 6 reports macro-F1 scores obtained using both raw segment features and Δ features.

Table 6

Audio-only Random Forest probes (macro F1): comparison between Stratified CV and Speaker CV, using raw segment features and delta features Δ (segment – baseline).

	Segment (raw)		Delta (Δ)	
	Stratified CV	Speaker CV	Stratified CV	Speaker CV
Subtask 1	0.742	0.529	0.553	0.511
Subtask 2	0.646	0.494	0.547	0.558
Subtask 3	0.568	0.391	0.405	0.324

The results highlight two complementary patterns. On the one hand, Δ features yield overall low performance, especially on subtask 3, indicating limited discriminative capacity with respect to pragmatic classes. Although they capture systematic local shifts relative to the baseline (e.g., short-term changes in voicing/pauses and energy), such variations are not sufficiently informative to robustly separate the target categories.

On the other hand, raw segment features achieve higher scores under stratified CV, but drop sharply under grouped CV, revealing substantial speaker leakage. The model tends to exploit speaker-idiosyncratic traits, such as timbre, prosodic range, articulation habits, rather than generalizable prosodic cues linked to the pragmatic phenomenon. Overall, these findings explain why, despite measurable local variations around triggers, the audio modality provides limited benefits for pragmatic classification and remains vulnerable to speaker-specific bias, thus motivating our final choice of a text-only approach, which outperforms the audio-only approaches (Table 6).

5.4. Analysis over the developed multimodal approaches

Before finalizing the proposed text-only architecture, we investigated several multimodal approaches to test the hypothesis that paralinguistic features could resolve pragmatic ambiguities.

The audio pipeline has been built on Wav2Vec 2.0 [28] in its large-xlsr-53-italian version, preferred over WavLM [29] for its better phonetic sensibility towards the Italian language. To implement the proposed Span-Aware approach, we isolated the acoustic component associated with the trigger using WhisperX [30] to implement a forced alignment at word-level. Additionally, we refined the temporal alignment through fuzzy string matching to compensate for transcription discrepancies.

The integration between local and global representation has been implemented with cross-attention. More in detail, two paradigms have been investigated:

1. **Cross-Modal Attention Network** we used a cross-attention mechanism in which textual representation from UmBERTo serves as Query, while audio feature extracted from Wav2Vec and enriched with prosodic descriptors, serves as Key and Value, as in Multimodal Transformer (MulT) [31]. This approach, similar to CM-BERT [32] and CCMT [33], dynamically manages the temporal misalignment.
2. **MTAMW Re-implementation** we implemented the Multimodal Transformer with Adaptive Modality Weighting (MTAMW) architecture [34], designed to adaptively weight the importance of both modalities. Unfortunately, obtained results were comparable to, or slightly lower than, those achieved with standard cross-attention.

Despite the architectural sophistication, empirical results demonstrated the ineffectiveness of the acoustic component for this specific task. A simple audio-only model based on a fine-tuned version of Wav2Vec 2.0 reached a macro F1-score of only 0.5917 on subtask 1. This result over the Stratified CV split, resulted significantly lower compared with text-only approaches, on the same split. On the other hand, the multimodal architecture involving both modalities and a cross-attention merging strategy, obtained a macro-F1 score of 0.9636 in subtask 1. Notwithstanding the promising results, the removal of the audio component at inference time, led to no significant changes in the final result.

These experiments showed the strength of the UmBERTo textual encoder in capturing the semantic context, and, by contrast, the additional contribution provided by the audio, results orthogonal but also redundant and noisy. The orthogonality is given by the diverse nature of information acquired and coded in audio signals compared with the textual ones, while redundancy and noisiness are provided by the alignment unreliability and speaker bias. Overall, despite their potentialities, audio features resulted to be not significantly discriminative: therefore, to maximize the computational efficiency and the robustness of the entire system, we opted for a text-only model (Section 2).

6. Conclusion

In this work we presented the proposed model by RES2 team for the IMPOLS challenge in the context of EVALITA 2026 campaign. The developed system won the competition for subtasks 2 and 3 and reached the 2nd place for the 1st subtask. The strengths of the proposed system rely on the Span-Awareness strategy, which combines both the target trigger with its context using UmBERTo to extract embeddings. The definition and usage of a Span-Preserving Data Augmentation with Gemini 3.0, allows us to increment the size of the training set, providing also more stability and robustness to the final trained model. In addition, the proposed pipeline confirms that, for some tasks in specialized domains, the usage of fine-tuned encoder-only models can overcome generalist models such as generative LLMs.

Moreover, the analysis on the audio data and its usage within the textual transcription in a multimodal setup, highlighted the presence of a severe speaker bias. In these cases, the extracted acoustic features guide the models towards the identification of the speaker, rather than focusing on prosodic features of the implicature. On doing this, the audio signal results redundant or noisy relative to the text, and its usage in a multimodal architecture is limited.

In future works we will focus on a better fine-grained audio-text alignment to isolate the local prosodic variation in correspondence to the triggers, as well as in developing an automatic trigger detection methods. Another interesting aspect will be the analysis of performance of generative LLMs, e.g. with the adoption of Retrieval-Augmented Generation (RAG) techniques. With this strategy,

it would be possible to enhance the context to provide to the model with encyclopedic information regarding speakers, political affiliations, and socio-historical backgrounds. This direction is particularly relevant when implicitness is not fully recoverable from a single lexical trigger, but emerges from broader contextual and world-knowledge cues typical of political discourse. Furthermore, we envision the implementation of fine-tuning based on reasoning paths to inject explicit pragmatic explanations that justify the nature of implicit content. We aim that this strategy may enhance logical inference capabilities of the model, along its precision in discriminating between complex linguistic categories.

Declaration on Generative AI

During the preparation of this work, the author(s) used Gemini 3 in order to: Grammar and spelling check, Text Translation. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

- [1] L. Gregori, W. Paci, V. Saccone, IMPOLS at EVALITA 2026: Overview of the IMPLICIT contents in POLITICAL Speech Task, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [2] F. Cutugno, A. Miaschi, A. P. Aproso, G. Rambelli, L. Siciliani, M. A. Stranisci, Evalita 2026: Overview of the 9th evaluation campaign of natural language processing and speech tools for Italian, in: *Proceedings of the Ninth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2026)*, CEUR.org, Bari, Italy, 2026.
- [3] G. Comandini, G. Ferrari, A. Panunzi, Impaqts: A multimodal corpus of implicit manipulatory strategies in political speech, in: *Proceedings of the Ninth Italian Conference on Computational Linguistics (CLiC-it 2023)*, CEUR-WS.org, Venice, Italy, 2023. URL: <https://ceur-ws.org/Vol-3596/>.
- [4] W. Paci, A. Panunzi, S. Pezzelle, They want to pretend not to understand: The limits of current llms in interpreting implicit content of political discourse, in: *Findings of the Association for Computational Linguistics: ACL 2025*, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 15569–15593. doi:10.18653/v1/2025.findings-acl.804.
- [5] L. Parisi, S. Francia, P. Magnani, Umberto: an Italian language model trained with whole word masking, <https://github.com/musixmatchresearch/umberto>, 2020.
- [6] Google DeepMind, Gemini 3.0: A new era of intelligence, 2025. URL: <https://deepmind.google/gemini>, technical Report. Retrieved from <https://deepmind.google/gemini>.
- [7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is All you Need, in: *Advances in Neural Information Processing Systems*, 2017.
- [8] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, O. Levy, SpanBERT: Improving pre-training by representing and predicting spans, *Transactions of the Association for Computational Linguistics* 8 (2020) 64–77. URL: <https://aclanthology.org/2020.tacl-1.5/>. doi:10.1162/tacl_a_00300.
- [9] L. Baldini Soares, N. FitzGerald, J. Ling, T. Kwiatkowski, Matching the blanks: Distributional similarity for relation learning, in: A. Korhonen, D. Traum, L. Màrquez (Eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Florence, Italy, 2019, pp. 2895–2905. URL: <https://aclanthology.org/P19-1279/>. doi:10.18653/v1/P19-1279.
- [10] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), 2023. URL: <https://arxiv.org/abs/1606.08415>. arXiv:1606.08415.
- [11] D. I. Beaver, B. Geurts, Presupposition, *Stanford Encyclopedia of Philosophy*, 2011. URL: <https://plato.stanford.edu/entries/presupposition/>, first published 2011; substantive revision 2021.
- [12] C. N. Silla, A. A. Freitas, A survey of hierarchical classification across different application domains, *Data Mining and Knowledge Discovery* 22 (2011) 31–72.

- [13] R. Caruana, Multitask learning, *Machine Learning* 28 (1997) 41–75.
- [14] L. Liebel, M. Körner, Auxiliary tasks in multi-task learning, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1–10.
- [15] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, et al., RoBERTa: A Robustly Optimized BERT Pretraining Approach, *arXiv preprint arXiv:1907.11692* (2019).
- [16] T. Kudo, J. Richardson, Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 66–71. doi:10.18653/v1/D18-2012.
- [17] T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, Optuna: A next-generation hyperparameter optimization framework, in: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, Anchorage, AK, USA, 2019, pp. 2623–2631. doi:10.1145/3292500.3330701.
- [18] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, et al., Pytorch: An imperative style, high-performance deep learning library, in: *Advances in Neural Information Processing Systems*, volume 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: <https://proceedings.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [19] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, et al., Qwen Technical Report, *arXiv preprint arXiv:2309.16609* (2023).
- [20] S. Schweter, Italian bert and electra models, 2020. URL: <https://github.com/dbmdz/berts>. doi:10.5281/zenodo.4271493.
- [21] P. He, J. Gao, W. Chen, Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing, in: *Eleventh International Conference on Learning Representations (ICLR)*, 2023. URL: <https://openreview.net/forum?id=sE7-XhLxHA>.
- [22] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, et al., Unsupervised cross-lingual representation learning at scale, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 8440–8451.
- [23] D. Atzeni, M. Guredda, A. Moretti, Gilberto: a pre-trained language model for italian, 2020. URL: <https://github.com/idb-ita/GilBERTo>.
- [24] P. Boersma, D. Weenink, Praat: doing phonetics by computer [computer program], 2024. URL: <http://www.praat.org/>, retrieved from <http://www.praat.org/>.
- [25] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, O. Nieto, librosa: Audio and music signal analysis in python, in: K. Huff, J. Bergstra (Eds.), *Proceedings of the 14th Python in Science Conference*, 2015, pp. 18–25. doi:10.25080/Majora-7b98e3ed-003.
- [26] F. Eyben, K. R. Scherer, B. W. Schuller, J. Sundberg, E. André, et al., The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing, *IEEE Transactions on Affective Computing* 7 (2016) 190–202. doi:10.1109/TAFFC.2015.2457417.
- [27] F. Eyben, M. Wöllmer, B. Schuller, opensmile: The munich versatile and fast open-source audio feature extractor, in: *Proceedings of the 18th ACM International Conference on Multimedia*, ACM, Firenze, Italy, 2010, pp. 1459–1462. doi:10.1145/1873951.1874246.
- [28] A. Baevski, Y. Zhou, A. Mohamed, M. Auli, wav2vec 2.0: A framework for self-supervised learning of speech representations, in: H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020, pp. 12449–12460. URL: <https://proceedings.neurips.cc/paper/2020/file/92d1e1eb1cd6f9fba3227870bb6d7f07-Paper.pdf>.
- [29] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, et al., Wavlm: Large-scale self-supervised pre-training for full stack speech processing, *IEEE Journal of Selected Topics in Signal Processing* 16 (2022) 1505–1518. doi:10.1109/JSTSP.2022.3188113.
- [30] M. Bain, J. Huh, T. Han, A. Zisserman, Whisperx: Time-accurate speech transcription of long-form audio, in: *INTERSPEECH 2023, ISCA*, 2023, pp. 148–152. URL: <https://doi.org/10.21437/Interspeech.2023-78>. doi:10.21437/Interspeech.2023-78.
- [31] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, R. Salakhutdinov, Multimodal trans-

- former for unaligned multimodal language sequences, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 6558–6569. doi:10.18653/v1/P19-1656.
- [32] K. Yang, H. Xu, K. Gao, Cm-bert: Cross-modal bert for text-audio sentiment analysis, in: Proceedings of the 28th ACM International Conference on Multimedia, ACM, Seattle, WA, USA, 2020, pp. 521–528. doi:10.1145/3394171.3413690.
- [33] N.-C. Ristea, A. Anghel, R. T. Ionescu, Cascaded cross-modal transformer for audio-textual classification, arXiv preprint arXiv:2401.07575 (2024). URL: <https://arxiv.org/abs/2401.07575>.
- [34] Y. Wang, J. He, D. Wang, Q. Wang, B. Wan, X. Luo, Multimodal transformer with adaptive modality weighting for multimodal sentiment analysis, Neurocomputing 572 (2024) 127181. doi:10.1016/j.neucom.2023.127181.

INDAQA2 - A Large Italian Narrative QA Benchmark: A CALAMITA 2026 Challenge

Luca Gioffré^{1,*}, Luca Moroni¹, Alberte Fernández-Castro^{1,2}, Elena Marafatto¹, Giacomo Garufi¹ and Roberto Navigli^{1,2}

¹Sapienza NLP Group, Dip. di Ingegneria Informatica, Automatica e Gestionale, Sapienza University of Rome, Italy

²Babelscape, Rome, Italy

Abstract

Long-context comprehension and reasoning remain largely underexplored in the evaluation of Italian Large Language Models (LLMs). Existing Italian benchmarks primarily focus on short or medium-length inputs, offering limited insight into models' ability to process extended narratives. To address this gap, we introduce INDAQA2, a substantially revised and expanded version of INDAQA, a benchmark for narrative question answering on original Italian literary texts. The new version comprises an expanded corpus of 461 total books, introduces a multiple-choice question answering format alongside the original open-ended tasks, and features manually curated texts drawn exclusively from works originally written in Italian, thus avoiding artifacts introduced by translation. The benchmark evaluates long-context understanding over complete books of up to 250K tokens, testing complementary comprehension skills through a dual-structure design: global narrative understanding, assessed via questions derived from book summaries, and local precision, assessed via questions grounded in specific passages and entity-level details. By supporting both open-ended and multiple-choice question answering formats, INDAQA2 enables evaluation of both generative capabilities and discriminative reasoning, facilitating comprehensive and scalable comparison across models. Our evaluation of several Italian-specialized and multilingual models reveals significant performance disparities across task formats and highlights limitations in how current Italian models utilize extended contexts.

Keywords

Narratives, Question-Answering, Long-context, Evaluation, Benchmark

1. Challenge: Introduction and Motivation

In recent years, an increasing number of newly released Large Language Models (LLMs) have been explicitly designed to process long inputs [1, 2]. Context windows have expanded rapidly, from early models limited to a few thousand tokens [3] to systems such as Llama 3.1 [4], supporting up to 128K tokens, and Qwen 2.5 [5] and Gemini 1.5 [6], which extend this capacity to the order of one million tokens. While these advances are impressive, the ability to handle long contexts introduces new and non-trivial challenges for model evaluation.

In parallel with the development of techniques to extend context length, there has been a surge in benchmarks targeting long-context capabilities. Prominent examples include Needle-in-a-Haystack-style tests [7, 8] and their long-context extension BABILong [9], which primarily assess retrieval and robustness under increasing input length. Beyond simple retrieval, benchmarks such as ∞ Bench [10] extend evaluation to contexts exceeding 100K tokens across diverse domains, while RULER [11] introduces multi-hop tracing and aggregation tasks to test deeper reasoning capabilities. However, most of these benchmarks are strongly English-centric, with only a limited number of multilingual efforts (such as BABILong-ITA [12]), and they rarely focus on deep, narrative-level reasoning.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

✉ gioffre@diag.uniroma1.it (L. Gioffré); moroni@diag.uniroma1.it (L. Moroni); castro@diag.uniroma1.it (A. Fernández-Castro); marafatto@diag.uniroma1.it (E. Marafatto); garufi.1750327@studenti.uniroma1.it (G. Garufi); navigli@diag.uniroma1.it (R. Navigli)

🌐 <https://lukfre.github.io/> (L. Gioffré)

🆔 0009-0007-9705-6797 (L. Gioffré); 0009-0006-1210-5098 (L. Moroni); 0000-0003-3831-9706 (R. Navigli)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Despite recent efforts to advance generative evaluation for Italian [13, 14, 15, 16], a language spoken by over 60 million native speakers and supported by a rich literary tradition, comprehensive benchmarks for evaluating both retrieval and reasoning in long-document abstractive question answering remain largely absent [17]. The absence of such resources hinders the development and evaluation of Italian language models on tasks requiring extended discourse comprehension, a capability increasingly important for applications ranging from document analysis to educational tools and accessibility technologies.

To the best of our knowledge, there exist only two datasets for narrative QA in Italian, but each has significant limitations. FairytaleQA-IT [18], a machine translation into Italian of the FairytaleQA benchmark [19], is limited to children’s fairy tales and may contain translation artifacts that do not reflect natural Italian language use. INDAQA [20], the first Italian long-context QA benchmark on narratives, adopts the NarrativeQA methodology of generating questions from human-written summaries rather than directly from source texts. While this approach ensures question quality, it may inadvertently constrain question types to those answerable from compressed versions of narratives, rather than questions requiring deep engagement with the full text. Moreover, the benchmark lacks a systematic quality check of the source texts, and by supporting only open-ended generative tasks, it requires a carefully tailored evaluation setup, both important factors for building a fair (narrative) benchmark [21, 22]. Supporting the multiple-choice task would allow for a more straightforward evaluation, despite its own structural problems [23].

To bridge this gap, we introduce an expanded and substantially refined version of INDAQA, which we name **INDAQ2**. The new benchmark addresses the aforementioned limitations and extends the dataset with an additional collection of 99 long texts paired with (open-ended and multiple-choice) question–answer items generated under different strategies, complementing the challenges and evaluation principles outlined in the CALAMITA initiative [16]. As in the original benchmark, all source texts come from gold Italian novels or theatrical screenplays, manually collected and curated. Importantly, the newly added works consist primarily of lesser-known literary texts, reducing the likelihood that models have memorized substantial amounts of content from their training data and thus mitigating contamination effects. Moreover, all texts are authored in Italian and written directly in the language, avoiding artifacts introduced by translation. In some cases, the texts also include regional varieties and dialects (e.g., works written by *Goldoni* or *Pirandello*), further increasing linguistic diversity. Together, these properties make the benchmark a more reliable and realistic test bed for evaluating long-context comprehension and reasoning in Italian LLMs. For the aforementioned reasons, we expect models to perform rather poorly on this new challenge, mainly due to the length of the inputs.

2. Challenge: Description

This challenge is focused on Open-ended (OE) and Multiple-choice (MC) QA over long, narrative texts. Thus, it supports two main tasks, depending on the expected output of the model (free-form, open-ended generation or choice labels).

We focus on narratives for two main reasons. First, **many narrative works are long by-design**, so we can avoid constructing a synthetic benchmark stitching together documents coming from different sources, as done in RULER [11] and BABILong [9]. The second reason is that **the narrative domain represents a critical challenge** for natural language understanding systems [21]. Unlike factoid QA tasks over short passages [24, 25], the narrative QA task requires models to maintain coherence across thousands of tokens, track character relationships, understand causal chains, and reason about plot developments.

All QA items in the benchmark have been automatically generated either from the summary of the story, following the style of NarrativeQA [26], or from individual or grouped excerpts (Section 3).

INDAQ2 - OEQA task The model is prompted with a simple instruction, followed by the entire content of the book and the question. Then, the model is expected to generate a concise answer, which will be evaluated against a set of reference answers. The number of reference answers, which spans

from 1 to 5, depends on the difficulty of answering the question: questions which may allow for different formulations of the correct answers (i.e., paraphrases) have more references, so that the generated answers are scored fairly.

INDAQA2 - MCQA task The model is prompted with a simple instruction, followed by the entire content of the book, the question, and four answer choices (three distractors and the correct answer). The choices are already shuffled, so that the correct answer can be found in any position with equal probability. The model is expected to identify the correct answer and return the letter (i.e., the label) corresponding to the option deemed correct. Since the correct answer is given as one of the four options, in this setting there is no need for a set of reference answers.

la_coscienza_di_zeno.oe.42 Chi è Carla per Zeno? - È la sua amante - È la donna con cui ha una relazione extraconiugale.	la_coscienza_di_zeno.mc.42 Chi è Carla per Zeno? A. È un'amica di sua moglie B. È una parente C. È la sua amante D. È la sua cameriera
---	--

Figure 1: Examples of a QA item from *La coscienza di Zeno*. In the left side is the OE item, for which both reference answers are true. In the right side, is the MC item: only one of the four options is correct (marked in bold).

3. Data description

Our goal is to create a large-scale question-answering benchmark grounded in Italian literary texts. Building upon the 362 books from the original INDAQA, we extend the collection with 99 additional works, resulting in a total of 461 books sourced from Project Gutenberg, Wikisource, and LiberLiber (Section 3.1). We also collect introductory sections as well as summary or plot sections from the corresponding Wikipedia pages when available. However, manually writing questions, reference answers, and distractors (collectively, QA items) at book scale is unfeasible. Therefore, we rely on an LLM to generate and refine QA items, following an approach similar to [20].

Our QA generation is grounded in different types of textual context, which leads to two data splits. In the **Summary-level** split, QA items are generated from book summaries available on the corresponding Wikipedia pages (Section 3.2). In the **Passage-level** split, QA items are instead generated from individual book passages, as no summaries are available for these works (Section 3.3).

To assess the quality and appropriateness of the generated data, we conduct a manual annotation study on a representative sample of QA items (Section 3.4).

The data format and the inference prompts used in our evaluation are described in Sections 3.5 and 3.6. We provide detailed statistics in Section 3.7 and Section C, while the prompts used for QA generation and refinement are available in Section A.

3.1. Origin of data

All documents in the benchmark have been downloaded from either Project Gutenberg, Wikisource or LiberLiber¹. We also use Wikipedia to extract summaries and metadata (when available, see Section B).

Project Gutenberg is a volunteer-driven initiative dedicated to digitizing and preserving cultural works and to encouraging the creation and distribution of e-books. Founded in 1971, it is the oldest existing digital library. Its collection consists mainly of complete books or individual texts in the public

¹<https://www.gutenberg.org/>, <https://it.wikisource.org/>, <https://liberliber.it/>

domain. All materials are freely accessible and provided in open, non-proprietary formats compatible with nearly all computing platforms.

Wikisource is a project of the Wikimedia Foundation that aims to build a freely accessible online library of original source texts and their translations in any language. Unlike Project Gutenberg, Wikisource documents can be collaboratively edited and reviewed by contributors, allowing continuous improvement, verification against original sources, and transparent version tracking. As a result, the quality and reliability of texts generally increase over time.

LiberLiber is an Italian non-profit project focused on promoting free culture through the publication of literary, musical, and scholarly works in the public domain or released under free licences. Its digital library places great emphasis on textual accuracy, careful proofreading, and scholarly reliability. Often, documents hosted on LiberLiber are considered to be of higher editorial quality than those available on Project Gutenberg.

3.2. Summary-level data split

Source texts The Summary-level data split contains 362 books for which a summary or plot section is available on Wikipedia, corresponding to the original INDAQA dataset [20]. During our analysis, we found that several books contained incomplete or corrupted content. For instance, many theatrical screenplays were missing character names in dialogues, while others exhibited missing or duplicated sections. These issues were likely caused by failures of the HTML parser when applied to heterogeneous, crowdsourced texts that lack a standardized structure. It is indeed difficult to devise a parsing method which accounts for different HTML formatting, as one effective for one document could produce corrupted outputs for others. To address this issue, we manually downloaded and reviewed the source texts for all documents, ensuring high quality across the dataset².

QA items generation Since the QA items were generated from the summaries, which we found to be of high quality, and manually validated in [20], we do not modify them.

The resulting Summary-level section of INDAQA2 contains the original summaries, questions and answers, with clean source books for each QA item. Figure 13 in Section D shows an example of a Summary-level QA item as a JSON object.

3.3. Passage-level data split

Source texts While valuable, the Summary-level data split mainly accounts for questions either about the whole narrative (*abstractive*) or about information for which the answer can be found in various passages. To test the ability of the models to retrieve more localized and specific details, we extend the original 362 documents with a collection of another 99 books that were discarded in the previous INDAQA release since they were lacking a summary. These books were downloaded from the same sources and followed the same approach of the aforementioned data split.

QA item generation Lacking a summary, we devised three methods for generating QA items with an LLM starting from individual passages. These methods categorize the questions into three sets:

1. **Local**: questions generated from a single passage (defined as 20 contiguous sentences) randomly selected at runtime (prompt in Figure 5). These questions typically focus on specific details stated in the provided passage.
2. **Local (alternative)**: questions generated from a single passage plus the previously generated *Local* items (prompt in Figure 6). We noticed that this generation setting encourages the model to avoid repetition and generate less straightforward questions, shifting the distribution of question types (see Figure 3).

²We downloaded the text directly in .txt or .rtf formats, avoiding HTML pages.

3. **Entity**: questions generated from three passages in which an entity consistently appears, selected from the beginning, middle, and ending sections of the documents (prompt in Figure 7). Entities are identified by extracting all capitalized names (excluding *stopwords*) from the questions in the two previously defined sets. We clustered the entities by a simple exact string match. These capitalized terms capture recurring entities such as characters, locations, or organizations that are central to the narrative. For each entity cluster, we select three passages where the entity appears. Entities with fewer than three occurrences or very low frequency are filtered out. The resulting clusters are manually validated to remove noise (e.g., common words erroneously capitalized, non-entities). Finally, the selected passages and their associated local questions are provided to the LLM to generate questions targeting overarching plot elements, character development, or thematic connections across the narrative.

MCQA conversion Due to their localized scope (i.e., a single passage), *Local* and *Local (alternative)* QA items are well suited to a multiple-choice format. For each item in these two sets, we provide an LLM with the source passage and the generated question with reference answers, and instruct it to produce three plausible distractors, preferably grounded in the given context (prompt in Figure 8). We also tried to convert the *Summary* and *Entity* questions to a multiple-choice format. However, we found that generating plausible distractors from summaries or multiple distant passages resulted in hallucinations or generally weak, low-quality distractors. Hence, only the two *Local* sets support the MCQA format.

QA items refinement Following the original methodology for the INDAQA dataset, we refine the *Local* QA items by asking an LLM to assess whether the questions are well posed and answerable, and whether the corresponding answers are acceptable. We exclude *Entity* questions from this correction step because their higher complexity could not be reliably handled by the tested LLMs. Figure 9 shows the prompt we used for this task, which we built following guidelines similar to those of [21]. The refinement process identified 49 QA items with problems: given their low number, we manually review the LLM correction and substitute the refined QA items in the benchmark.

Throughout our experiments, we used Gemini-2.5-Flash [27] to generate, refine and convert QA items. An example QA item for each question set can be found in Section D, Figures 14 to 16.

3.4. Annotation details

To assess the quality and validity of the generated QA items, we conduct human validation on a representative sample of the corpus. We focus on the newly added documents³, following similar annotation guidelines as in the original INDAQA [20].

Subset Selection The Passage-level split comprises 11,560 QA items across 99 books, of which 10,187 support both free-form and multiple-choice evaluation (the two *Local* question sets), while the remainder consists of only open-ended questions (the *Entity* question set). We target approximately 5% of the total dataset for human annotation. We adopt a stratified sampling strategy to select the books in the annotation set. Books are divided into 20 equal-probability bins based on text length quantiles, and from each bin, the book whose length is closest to that bin’s mean length is selected (Figure 11). We focus on 20 bins (and so, 20 books) to ensure manageable scope while maintaining diversity. This approach ensures representative coverage across the entire range of text lengths, avoiding potential biases.

Then, we randomly sample the QA items from the set of 20 books to mirror the distribution of the three question sets: 400 items from *Local* set (20 items per book), 120 items from *Local alternative* set (6

³The original INDAQA has an error rate of the generated QA items of 2.32%, which we deem acceptable.

items per book), and 60 items from *Entity* set (3 items per book), yielding a total of 580 elements to review.

To measure inter-annotator agreement (IAA), we randomly select 100 overlapping items from the annotation subset (~17%). This overlap also mirrors the overall QA item distribution: approximately 70% from the *Local* set, 20% from the *Local (alternative)* set, and 10% from the *Entity* set.

Annotation Guidelines The items in the annotation sample were independently validated by two expert annotators (either native or proficient in Italian). Annotators were asked to assess the quality of automatically generated QA items for Italian long-context narratives. For each of them, annotators evaluated the following dimensions:

- **Fluency:** Whether the question, correct answers, and eventual distractors are grammatically correct and naturally phrased in Italian.
- **Validity:** Whether the item elements are appropriate and accurate:
 - The question is clear and answerable given the source text
 - The reference answers are factually correct
 - The distractors are plausible but incorrect (if present)

Annotators were encouraged to note any unusual patterns, ambiguities, or issues worthy of further investigation.

Result After the annotation phase, we observed high inter-annotator agreement, with a Cohen’s Kappa of 0.7563. The average error rate in the dataset, defined as the proportion of non-acceptable items among the annotated sample, is 4.74%. From the annotation process, we observed that the generated questions were always well-posed and answerable from the provided context alone. The errors focused mostly on the reference answers, in particular on the second reference, which was sometimes inaccurate or wrong, while the first was always correct. Regarding the distractors, they were generally plausible, albeit sometimes weak. We discuss the implications of these quality observations for our evaluation methodology in Section 6.

3.5. Data format

The benchmark is freely available through the Hugging Face repository⁴. All items have the same data fields, but depending on the data section and question set, some may be empty.

- **id** (str): unique identifier for the document
- **text** (str): text of the document
- **qas** (list[dict]): QA entries associated with the document
 - **question_id** (str): unique ID for the QA item
 - **question** (str): the question text
 - **answers** (list): list of free-form reference answers
 - **choices** (list): list of MCQA options
 - **target** (dict):
 - * **label** (str): correct MCQ label ('A', 'B', 'C', or 'D')
 - * **text** (str): canonical correct answer (i.e., the first reference)
 - **entity** (str): entity targeted by the question, if present
 - **model** (str): generator model used
 - **kind** (str): question type (*Local*, *Local Alternative* or *Entity*)
 - **source_paragraphs_ids** (list): list of paragraph indices used to generate the QA
 - **source_questions_ids** (list): list of related question indices
- **metadata** (dict): book-level metadata

⁴https://huggingface.co/datasets/sapienzanlp/INDAQA_CALAMITA

Metric	Summary-level set	Passage-level set	Both sets
# Documents	362	99	461
# QA items	13,661	11,560	25,221
# QA items/doc	38 ± 2	117 ± 20	55 ± 34
Text length average	$26K \pm 33K$	$58K \pm 31K$	$33K \pm 35K$
Text length range	0.5K-242K	8K-188K	0.5K-242K

Table 1

Statistics for the documents in INDAQA2 divided by data split.

- **title** (str) : title of the work
- **author** (str) : author name
- **year** (int) : publication year
- **summary** (str) : book summary used in summary_level
- **summary_length** (int) : length of the summary (in words)
- **text_length** (int) : length of text (in words).
- **source_link** (str) : link to the text source
- **summary_link** (str) : link to the summary source
- **qa_paragraphs** (list[str]) : list of text chunks used to generate the QAs

We also report one example per data split and question set in Appendix Section D, Figures 13 to 16.

3.6. Example of prompts used for zero or/and few shots

The challenge is supposed to be accomplished in a zero-shot setting with a very simple prompt. We show in Figure 4 the prompt used for inference (in the generative task, `choices_block` is null).

3.7. Detailed data statistics

Tables 1 and 2 report statistics for the Summary- and Passage-level splits of the INDAQA2 dataset. Specifically, Table 1 presents document-level statistics divided by data split, including the number of documents, the average number of QA items per document, and text length. Table 2, instead, summarizes the distribution of question types (Summary, Local, and Entity) across the dataset. In Figure 2 we plot the length distribution of both data splits. Additionally, in Figure 3 we plot the question categorization (by first word) distribution across question types. Further statistics are presented in Section C.

Summary-level This split comprises 13,661 open-ended QA items spanning 362 documents, averaging approximately 38 QA items per book. Due to the cleaning process applied to the original INDAQA dataset, document lengths were reduced by about ~3% while preserving all narrative content, from an average of $27K \pm 38K$ words to an average of $26K \pm 33K$ words.

Passage-level This split comprises 11,560 open-ended QA items spanning 99 documents. While containing fewer documents, it still has a comparable number of QA items with respect to the Summary-level split through a higher density of QA items. Notably, Passage-level documents are more than double the length of *Summary-level* ones on average, requiring models to process and reason over more extensive textual contexts.

4. Metrics

We employ different evaluation methodologies for the two settings in INDAQA2: *n*-gram-based metric for generative questions and accuracy for multiple-choice questions. In practice, as with all CALAMITA challenges, evaluation is carried out using the LM Evaluation Harness framework developed by EleutherAI.⁵

⁵<https://github.com/EleutherAI/lm-evaluation-harness>

Question Type	# Items	# Items/doc	Question length	Answer length
<i>Summary-level split</i>				
Summary	13,661 (100%)	38 ± 2	7 ± 2	5 ± 3
<i>Passage-level split</i>				
Local	7,901 (68%)	80 ± 14	8 ± 2	4 ± 2
Local (alternative)	2,286 (20%)	23 ± 5	9 ± 3	6 ± 4
Entity	1,373 (12%)	14 ± 6	13 ± 3	24 ± 8

Table 2

QA item distribution and length statistics by question type. Entity Questions feature notably longer answers (average 24 words vs. 4–6 words for other types), while Local Questions dominate the Passage-level set (68%). Question and answer lengths are measured in words; percentages represent proportions within each set.

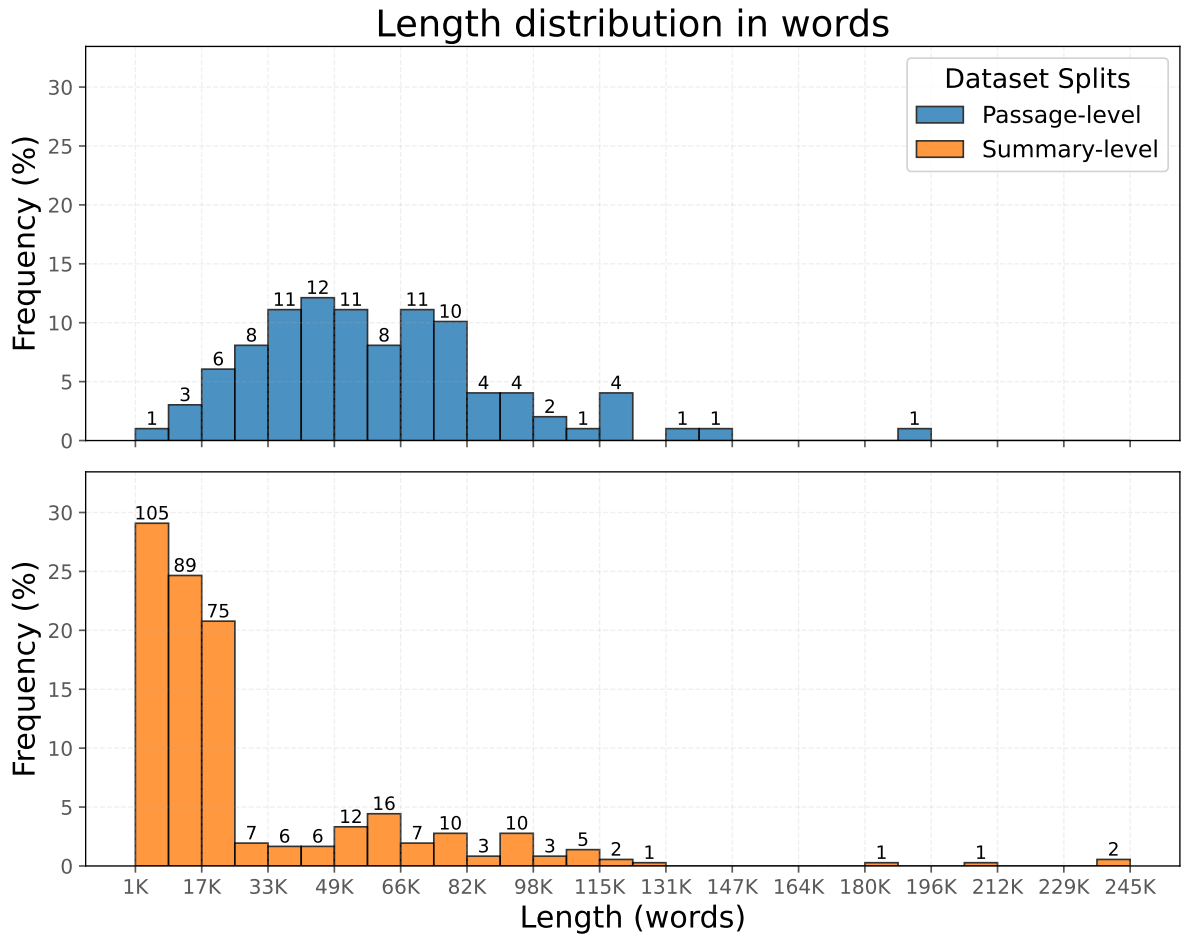


Figure 2: Document Length distribution across the two splits of INDAQA2. While the distribution of the original dataset is skewed towards shorter documents, the new data split is more balanced.

OEQA Evaluation For the generative task of INDAQA2, we employ **Exact-Match** (EM) and **METEOR** [28, 29] as automatic evaluation metrics. For both metrics, we take the maximum score between the candidate answer and the set of reference answers as the score of the model for that QA item.

EM is a simple and intuitive measure that checks whether a reference answer appears verbatim within the generated output. However, this metric may fail to capture semantically correct responses that do not exhibit lexical overlap with the reference. To address these well-known limitations of EM, we additionally adopt METEOR.

METEOR computes an alignment-based score between generated and reference answers by considering exact, stem, synonym, and paraphrase matches. Unlike simpler n -gram-based metrics such as ROUGE, METEOR jointly

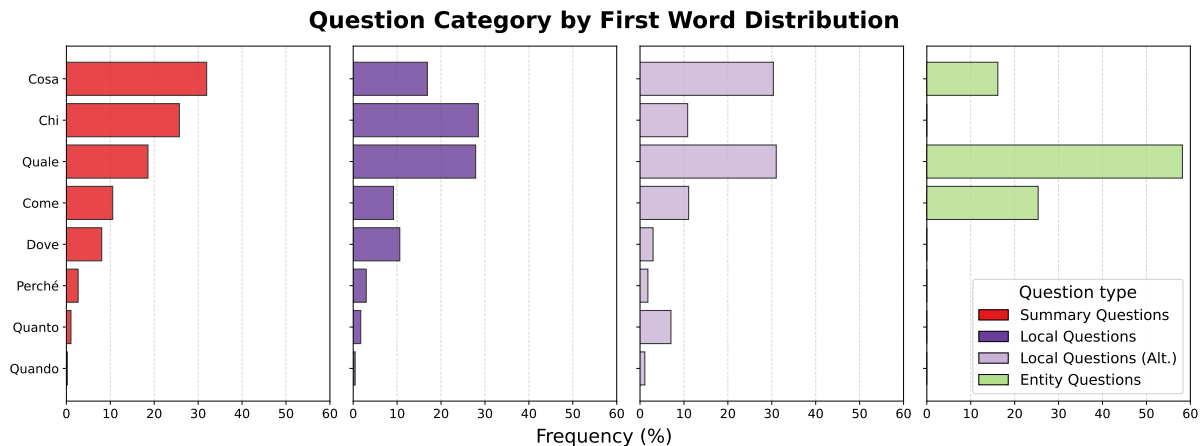


Figure 3: Distribution of first words of the questions. Each set has a different percentage of questions starting with a given first word.

models precision and recall while leveraging linguistic knowledge (e.g., synonyms) from WordNet, making it particularly suitable for evaluating natural language answers with different but equivalent surface forms. While there is not a standard metric for the evaluation of open-ended QA tasks, we follow prior work that demonstrates METEOR’s superior reliability in this domain [21].

MCQA Evaluation For the MCQA component of INDAQA2, we follow best practices for multiple-choice evaluation proposed in recent works, in which model outputs are evaluated using manually curated regular expressions [23, 30]. The set of regex patterns is constructed by inspecting the outputs of a subset of models and identifying the most frequent surface forms used to indicate the selected option. This approach follows the analysis and setup of [23], which shows that regex-based answer extraction is more reliable than perplexity-based evaluation pipelines that rely on next-token probabilities over candidate choices. Accuracy is computed as the proportion of questions for which an extracted pattern corresponds to the correct answer. The complete list of regex patterns is provided in Appendix E.

Context Truncation INDAQA2 contains documents of highly variable length, reaching up to hundreds of thousands of words (see Figure 2), which make the benchmark both memory- and computation-intensive. To have an affordable evaluation, we define multiple context-size settings by truncating each book to its first words⁶. Specifically, besides evaluating on the full book, we also test the models under two context lengths: 10K and 50K words. This setup allows us to assess system performance not only under partial but manageable contexts, but also in terms of efficiency in word splitting. Models that require fewer splits (i.e., having a lower fertility) are better suited to incorporate partial or full contextual information within their context window.

5. Results, Challenge-Specific Insights and Lessons Learned

We analyze and report the results of the INDAQA2 challenge, examining the performance of LLMs featured on the CALAMITA leaderboard. Our analysis is complemented by a qualitative discussion of a sample of generated outputs, as well as a critical assessment of current limitations and future challenges in Italian narrative understanding.

Performance across models (types, sizes) Table 5 reports the performance of the five instruction-tuned models evaluated in the INDAQA2 challenge. Specifically, we consider two Llama models—Llama3.1-8B and Llama3.1-70B [4]—the ANITA-8B model [31], and two variants of Minerva: Minerva-7B [32] and Minerva-7B_L [20]. Minerva-7B_L is an extension of Minerva-7B that underwent continual training to increase the maximum context length from 4K to 32K tokens. By comparison, ANITA-8B natively supports an 8K token context window, while both Llama models handle input sequences of up to 128K tokens.

⁶We simply split on whitespace.

QA item	Model answers	METEOR
Question Dove stava andando la marchesa?	<i>ANITA-8B</i> : Non andava da nessuna parte, stava seduta in un salotto di Bologna.	0.31
References – A Bologna – Verso la città di Bologna	<i>Llama-8B</i> : La marchesa stava andando a pranzo da Giorgio, il conte De Vinci, e poi avrebbe accompagnato Mimy, la moglie di Carlo, a pranzo da lei. <i>Minerva-7B_L</i> : La marchesa stava andando a Bologna.	0.13 0.89

Table 3

Model responses on a Local question in the OE task, paired with their METEOR score.

Among the 7-8B models, performance patterns vary considerably across context sizes and task types. In the 10K-word context setting, Llama3.1-8B achieves the strongest results on the *Local* and *Local alternative* multiple-choice tasks, reaching 43.83% and 43.15% accuracy respectively. Interestingly, the pattern reverses for the more challenging open-ended generation tasks: Minerva-7B_L outperforms all other 7-8B models on the *Summary*, *Local*, and *Entity* sets when measured by METEOR and EM scores. This suggests that Italian-specialized models may retain an advantage for generative tasks requiring nuanced language production, even when the multilingual model are better at discriminative multiple-choice selection. Notably, Minerva-7B_L consistently surpasses both the English-centric Llama models and the other Italian-specific alternatives (ANITA-8B and the previous version Minerva-7B) on these open-ended tasks.

When context is extended to 50K words, Llama3.1-8B and Llama3.1-70B show large improvements on all the tasks, due to their capabilities in handling longer contexts. This is especially evident in the MC task, with an increment of more than 20 percentage points. In contrast, the corresponding METEOR scores for the open-ended *Local* set increase by only 12 points. This pattern suggests that Llama models are particularly adept at leveraging extended context for multiple-choice tasks, where the presence of answer options provides additional grounding that helps the model locate and select the correct response. For open-ended generation, however, the benefits of longer context are more modest: while the model may identify relevant information, accurately producing well-formed Italian answers remains challenging.

Italian-specialized models show minimal improvement on 50K words and full book settings across both task formats, with ANITA and Minerva variants demonstrating performance nearly identical to their 10K results. This stagnation is due to their limited context window with respect to Llama models.

With full book context available, Llama3.1-8B maintains its strong performance while showing only incremental gains over the 50K setting. This plateau suggests that 50K words may already encompass most information relevant to answering the benchmark questions, with additional context providing diminishing returns.

As expected, Llama3.1-70B substantially outperforms all smaller models across all metrics and context settings, with performance gains becoming more pronounced as context length increases. The model’s higher parameter count enables it to better exploit the additional information provided by longer contexts. Due to computational constraints, we were unable to evaluate Llama3.1-70B on the full book setting.

Error Analysis To further investigate the task-specific performance patterns observed in the quantitative results, we conduct a qualitative error analysis on selected examples from both evaluation settings. Table 3 illustrates model outputs for question from the *Local* set in the OE task. The results demonstrate that, in this example, METEOR scores effectively capture answer quality: Minerva-7B_L produces a more general but correct response, achieving a higher METEOR score. In contrast, ANITA and Llama-8B generate less accurate or incomplete answers, reflected in their lower scores. This example corroborates our earlier finding that Italian-specialized models maintain advantages in open-ended generation tasks when they can effectively access relevant context.

Table 4 presents a particularly revealing case where Minerva-7B_L is prompted with the same question under both OE and MC formats. While the model successfully generates the correct answer in the OE setting, it selects an incorrect option in the MC setting. This failure mode is consistent with the quantitative patterns discussed above: Minerva-7B_L demonstrates good performance on open-ended tasks but struggles with multiple-choice selection. The contrast highlights a fundamental difference in how these evaluation formats probe model capabilities, a finding that aligns with observations by [20]. While multiple-choice tasks primarily assess whether models can leverage answer options to discriminate between alternatives (a capability where Llama models excel), OE tasks require models to independently generate well-formed responses in the target language, where Italian-specialized models show relative strength.

QA item	Model answers	Score
Question Da quale città provenivano le commissioni per estirpare l'eresia valdese?	<i>OE setting</i> Da Torino	METEOR: 1.00
Choices A. Da Carignano B. Da Ginevra C. Dalla Francia D. Da Torino	<i>MC setting</i> B. Da Ginevra	Correct: False

Table 4

Comparison of the Minerva-7B_L responses on a Local question in the OE task and MC task, paired with their evaluation. In the OE task, the model correctly responds, while in the MC task, it is confounded by the four options and chooses the wrong one. The choice in **bold** is the correct response, which corresponds to the one of the reference answers in the OE task.

Model	Summary set		Local set			Local alt. set			Entity set
	EM	METEOR	EM	METEOR	ACC	EM	METEOR	ACC	METEOR
<i>10K-word context</i>									
ANITA-8B	3.92	23.12	10.38	14.92	38.08	4.16	11.47	36.37	20.15
Minerva-7B	3.24	25.37	8.15	16.98	28.57	2.89	12.56	25.08	21.56
Minerva-7B _L	5.08	26.93	11.87	20.31	33.52	6.08	15.31	27.48	22.21
Llama3.1-8B	4.46	25.60	14.44	18.66	43.83	9.89	16.65	43.15	20.83
Llama3.1-70B	3.41	<u>27.62</u>	<u>17.04</u>	<u>23.24</u>	<u>49.57</u>	<u>11.72</u>	<u>19.71</u>	<u>50.02</u>	20.01
<i>50K-word context</i>									
ANITA-8B	4.26	24.06	11.37	15.36	39.46	4.86	11.79	37.64	20.57
Minerva-7B	3.45	25.81	8.37	17.02	29.91	2.97	12.64	25.95	21.46
Minerva-7B _L	5.30	27.80	13.10	21.05	33.91	5.99	15.07	26.43	22.17
Llama3.1-8B	5.45	28.42	29.44	30.60	64.98	26.20	34.16	66.91	22.87
Llama3.1-70B	3.21	<u>29.38</u>	<u>36.22</u>	<u>41.75</u>	<u>78.80</u>	<u>32.94</u>	<u>45.23</u>	<u>80.17</u>	<u>24.35</u>
<i>Full book context</i>									
ANITA-8B	4.21	25.38	10.99	15.51	39.50	4.72	12.04	37.68	19.55
Minerva-7B	3.61	26.28	7.68	16.71	30.42	2.62	12.34	26.12	21.31
Minerva-7B _L	5.42	27.94	12.94	20.75	33.54	5.90	15.11	26.47	22.10
Llama3.1-8B	5.51	30.74	31.73	34.24	67.77	28.39	37.87	70.15	22.77
Llama3.1-70B	-	-	-	-	-	-	-	-	-

Table 5

Results of four models on INDAQA2. We highlight in **bold** the best result per context setting on 7-8B models, and we underline the best result considering also the 70B model. Due to the high length of the responses in the Entity set, we do not show EM scores, as they are close to zero.

Critical analysis/discussion/future Our results carry important implications for the development of Italian language models and long-context evaluation more broadly. Current Italian-specialized models struggle with extended context utilization, suggesting that architectural innovations beyond continued pretraining may be necessary. Future work should explore whether techniques such as modified attention mechanisms, retrieval-augmented approaches, or different positional encoding schemes can better enable smaller Italian models to leverage long contexts. Moreover, the disparity between MC and OE performance raises questions about evaluation methodology: while both formats provide valuable signals, developing evaluation frameworks that better disentangle retrieval capabilities from generation quality could yield more actionable insights for model development.

6. Limitations

Historical and Linguistic Context In order to be copyright-free, INDAQA2 contains Italian literary texts spanning the period 1827-1948. As such, the language employed in these texts reflects the lexical and grammatical conventions of 19th and early 20th century Italian, including archaic vocabulary and syntactic structures that may differ substantially from contemporary usage. Additionally, the vast majority of authors in the corpus are males, which mirror the gender disparities inherent in that period. Consequently, the narrative content may embody attitudes, ideologies, and perspectives that reflect the sociocultural aspects of the period, some of which diverge from contemporary ethical and social standards.

Synthetic Data Although the QA items underwent quality control procedures, it is important to note that they were generated using an LLM. As such, the dataset may exhibit certain limitations inherent to LLM-generated content, including potential factual inaccuracies and systematic biases that reflect the model’s training data and architectural characteristics. Our annotation process (Section 3.4) revealed that while questions were consistently well-posed and answerable, approximately 4.74% of items contained errors, primarily in secondary reference answers. Since our evaluation setup (Section 4) computes the maximum of EM or METEOR scores across all references, the presence of at least one correct reference ensures valid answers are properly credited. However, inaccurate secondary references could theoretically reward incorrect model responses that happen to align with those errors rather than the correct answer. Given the low error rate and the fact that such biases manifest *only* when a model’s output matches the wrong reference instead of the correct one, we expect the practical impact on benchmark results to be limited. Nevertheless, future work could involve manual correction of identified errors to further enhance dataset quality.

Evaluation Setup While METEOR provides a reasonable automated evaluation framework for our task, we acknowledge its limitations. Automatic metrics based on surface form similarity may not fully capture semantic correctness, especially for questions requiring reasoning or inference. An LLM-as-a-Judge framework, where a powerful language model evaluates answer quality according to rubrics, has been shown to yield scores more aligned with human judgments in recent work [21, 33]. However, such approaches introduce additional computational costs, potential biases from the judge model, and complications in reproducibility. For the purposes of establishing baseline performance and enabling rapid iteration, we consider METEOR a pragmatic choice that balances evaluation quality with practical constraints.

7. Ethical issues

Given the publication years of the dataset (1827-1948), the source texts reflect the historical and sociocultural context of their time and may contain biased, stereotypical, or otherwise sensitive portrayals related to gender, ethnicity, religion, violence, or other forms of toxicity. While these aspects are intrinsic to the literary material and are preserved for research fidelity, their uncritical use may lead to the reproduction or amplification of outdated viewpoints in downstream applications.

We do not attempt to modify or filter such content, and we encourage users to interpret the dataset within its historical context and not to train models nor deploy models trained on it. The dataset is intended **exclusively** for research and benchmarking purposes.

8. Data licence and copyright issues

The dataset is constructed from publicly available data sources. The synthetic QA items do not have a copyright, as they respect the provider usage policy⁷. We will release the dataset upon acceptance for research use. All original copyrights remain with the respective content owners, and the dataset does not redistribute proprietary or restricted material. Users of the dataset are responsible for ensuring compliance with the licences of the original sources when using the data.

Acknowledgments

Luca Gioffré, Luca Moroni, and Alberte Fernández-Castro gratefully acknowledge the support of AI Factory IT4LIA project and the CINECA support for access to high-performance computing facilities. Elena Marafatto

⁷<https://policies.google.com/terms/generative-ai/use-policy>

gratefully acknowledges the support of Agenzia per la Cybersicurezza Nazionale. Roberto Navigli acknowledges the support of PNRR MUR project PE0000013-FAIR.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Claude in order to: Grammar and spelling check, Formatting assistance, Improve writing style. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] J. Su, Y. Lu, S. Pan, B. Wen, Y. Liu, Roformer: Enhanced transformer with rotary position embedding, ArXiv abs/2104.09864 (2021). URL: <https://api.semanticscholar.org/CorpusID:233307138>.
- [2] B. Peng, J. Quesnelle, H. Fan, E. Shippole, YaRN: Efficient context window extension of large language models, in: The Twelfth International Conference on Learning Representations, 2024. URL: <https://openreview.net/forum?id=wHBfxhZu1u>.
- [3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, OpenAI (2019). URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, accessed: 2024-11-15.
- [4] A. Grattafiori, the Llama 3 Team, The Llama 3 Herd of Models, 2024. URL: <https://arxiv.org/abs/2407.21783>. arXiv:2407.21783.
- [5] A. Yang, B. Yu, C. Li, D. Liu, F. Huang, H. Huang, J. Jiang, J. Tu, J. Zhang, J. Zhou, J. Lin, K. Dang, K. Yang, L. Yu, M. Li, M. Sun, Q. Zhu, R. Men, T. He, W. Xu, W. Yin, W. Yu, X. Qiu, X. Ren, X. Yang, Y. Li, Z. Xu, Z.-Y. Zhang, Qwen2.5-1m technical report, ArXiv abs/2501.15383 (2025). URL: <https://api.semanticscholar.org/CorpusID:275921951>.
- [6] G. Team, Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. URL: <https://arxiv.org/abs/2403.05530>. arXiv:2403.05530.
- [7] G. Kamradt, Needle in a haystack - pressure testing llms, 2023. URL: https://github.com/gkamradt/LLMTest_NeedleInAHaystack.
- [8] Y. Kuratov, A. Bulatov, P. Anokhin, D. Sorokin, A. Sorokin, M. Burtsev, In search of needles in a 10m haystack: Recurrent memory finds what llms miss, 2024. arXiv:2402.10790.
- [9] Y. Kuratov, A. Bulatov, P. Anokhin, I. Rodkin, D. Sorokin, A. Sorokin, M. Burtsev, Babilong: Testing the limits of llms with long context reasoning-in-a-haystack, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, volume 37, Curran Associates, Inc., 2024, pp. 106519–106554. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/c0d62e70dbc659cc9bd44cbcf1cb652f-Paper-Datasets_and_Benchmarks_Track.pdf.
- [10] X. Zhang, Y. Chen, S. Hu, Z. Xu, J. Chen, M. K. Hao, X. Han, Z. L. Thai, S. Wang, Z. Liu, M. Sun, ∞ bench: Extending long context evaluation beyond 100k tokens, 2024. URL: <https://arxiv.org/abs/2402.13718>. arXiv:2402.13718.
- [11] C.-P. Hsieh, S. Sun, S. Krizan, S. Acharya, D. Rekes, F. Jia, B. Ginsburg, RULER: What's the Real Context Size of Your Long-Context Language Models?, in: First Conference on Language Modeling, 2024. URL: <https://openreview.net/forum?id=kIoBbc76Sy>.
- [12] F. Tamburini, BABILong-ITA: A new benchmark for testing large language models effective context length and a context extension method, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 1112–1120. URL: <https://aclanthology.org/2025.clicit-1.105/>.
- [13] L. Moroni, S. Conia, F. Martelli, R. Navigli, Towards a more comprehensive evaluation for Italian LLMs, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 584–599. URL: <https://aclanthology.org/2024.clicit-1.67/>.
- [14] B. Magnini, M. Madeddu, M. Resta, R. Zanolli, M. Cimmino, P. Albano, V. Patti, A leaderboard for benchmarking LLMs on Italian, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 636–646. URL: <https://aclanthology.org/2025.clicit-1.61/>.
- [15] G. Puccetti, M. Cassese, A. Esuli, The invalsi benchmarks: measuring the linguistic and mathematical understanding of large language models in Italian, in: O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, S. Schockaert (Eds.), Proceedings of the 31st International Conference on Computational

- Linguistics, Association for Computational Linguistics, Abu Dhabi, UAE, 2025, pp. 6782–6797. URL: <https://aclanthology.org/2025.coling-main.453/>.
- [16] M. Nissim, D. Croce, V. Patti, P. Basile, G. Attanasio, E. Musacchio, M. Rinaldi, F. Borazio, M. Francis, J. Gili, D. Scalena, B. Altuna, E. Azurmendi, V. Basile, L. Bentivogli, A. Bisazza, M. Bolognesi, D. Brunato, T. Caselli, S. Casola, M. Cassese, M. Cettolo, C. Collacciani, L. D. Cosmo, M. P. D. Buono, A. Esuli, J. Etxaniz, C. Ferrando, A. Fidelangeli, S. Frenda, A. Fusco, M. Gaido, A. Galassi, F. Galli, L. Giordano, M. Goffetti, I. Gonzalez-Dios, L. Gregori, G. Grundler, S. Iannaccone, C. Jiang, M. L. Quatra, F. Lagioia, S. M. Lo, M. Madeddu, B. Magnini, R. Manna, F. Mercorio, P. Merlo, A. Muti, V. Nastase, M. Negri, D. Onorati, E. Palmieri, S. Papi, L. Passaro, G. Pensa, A. Piergentili, D. Poterì, G. Puccetti, F. Ranaldi, L. Ranaldi, A. A. Ravelli, M. Rosola, E. S. Ruzzetti, G. Samo, A. Santilli, P. Santin, G. Sarti, G. Sartor, B. Savoldi, A. Serino, A. Seveso, L. Siciliani, P. Torroni, R. Varvara, A. Zaninello, A. Zanollo, F. M. Zanzotto, K. Zeinalipour, A. Zugarini, Challenging the abilities of large language models in italian: a community initiative, 2025. URL: <https://arxiv.org/abs/2512.04759>. arXiv:2512.04759.
 - [17] L. Moroni, G. Pappacoda, E. Barba, S. Conia, A. Galassi, B. Magnini, R. Navigli, P. Torroni, R. Zanoli, Sustainable Italian LLM evaluation: Community perspectives and methodological guidelines, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 747–759. URL: <https://aclanthology.org/2025.clicit-1.71/>.
 - [18] B. Leite, T. F. Osório, H. L. Cardoso, FairytaleQA Translated: Enabling Educational Question and Answer Generation in Less-Resourced Languages, Springer Nature Switzerland, 2024, p. 222–236. URL: http://dx.doi.org/10.1007/978-3-031-72315-5_16. doi:10.1007/978-3-031-72315-5_16.
 - [19] Y. Xu, D. Wang, M. Yu, D. Ritchie, B. Yao, T. Wu, Z. Zhang, T. J.-J. Li, N. Bradford, B. Sun, T. B. Hoang, Y. Sang, Y. Hou, X. Ma, D. Yang, N. Peng, Z. Yu, M. Warschauer, Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 447–460. URL: <https://aclanthology.org/2022.acl-long.34/>. doi:10.18653/v1/2022.acl-long.34.
 - [20] L. Moroni, T. Bonomo, L. Gioffré, L. Xu, D. Fedele, L. Colosi, A. S. Bejgu, A. Scirè, R. Navigli, What we learned from continually training minerva: A case study on Italian, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 760–774. URL: <https://aclanthology.org/2025.clicit-1.72/>.
 - [21] T. Bonomo, L. Gioffré, R. Navigli, LiteraryQA: Towards effective evaluation of long-document narrative QA, in: C. Christodoulopoulos, T. Chakraborty, C. Rose, V. Peng (Eds.), Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Suzhou, China, 2025, pp. 34086–34107. URL: <https://aclanthology.org/2025.emnlp-main.1729/>. doi:10.18653/v1/2025.emnlp-main.1729.
 - [22] S. Tedeschi, J. Bos, T. Declerck, J. Hajič, D. Herscovich, E. Hovy, A. Koller, S. Krek, S. Schockaert, R. Sennrich, E. Shutova, R. Navigli, What’s the meaning of superhuman performance in today’s NLU?, in: A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Toronto, Canada, 2023, pp. 12471–12491. URL: <https://aclanthology.org/2023.acl-long.697/>. doi:10.18653/v1/2023.acl-long.697.
 - [23] F. M. Molfese, L. Moroni, L. Gioffré, A. Scirè, S. Conia, R. Navigli, Right answer, wrong score: Uncovering the inconsistencies of LLM evaluation in multiple-choice question answering, in: W. Che, J. Nabende, E. Shutova, M. T. Pilehvar (Eds.), Findings of the Association for Computational Linguistics: ACL 2025, Association for Computational Linguistics, Vienna, Austria, 2025, pp. 18477–18494. URL: <https://aclanthology.org/2025.findings-acl.950/>. doi:10.18653/v1/2025.findings-acl.950.
 - [24] M. Joshi, E. Choi, D. Weld, L. Zettlemoyer, TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension, in: R. Barzilay, M.-Y. Kan (Eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Vancouver, Canada, 2017, pp. 1601–1611. URL: <https://aclanthology.org/P17-1147/>. doi:10.18653/v1/P17-1147.
 - [25] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, C. D. Manning, HotpotQA: A dataset for diverse, explainable multi-hop question answering, in: E. Riloff, D. Chiang, J. Hockenmaier, J. Tsujii (Eds.), Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2369–2380. URL: <https://aclanthology.org/D18-1259/>. doi:10.18653/v1/D18-1259.
 - [26] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, E. Grefenstette, The narrativeqa

- reading comprehension challenge, 2017. URL: <https://arxiv.org/abs/1712.07040>. arXiv:1712.07040.
- [27] G. Comanici, the Gemini Team, Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities, 2025. URL: <https://arxiv.org/abs/2507.06261>. arXiv:2507.06261.
- [28] S. Banerjee, A. Lavie, METEOR: An automatic metric for MT evaluation with improved correlation with human judgments, in: J. Goldstein, A. Lavie, C.-Y. Lin, C. Voss (Eds.), Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 65–72. URL: <https://aclanthology.org/W05-0909/>.
- [29] A. Lavie, A. Agarwal, METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments, in: C. Callison-Burch, P. Koehn, C. S. Fordyce, C. Monz (Eds.), Proceedings of the Second Workshop on Statistical Machine Translation, Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 228–231. URL: <https://aclanthology.org/W07-0734/>.
- [30] X. Wang, B. Ma, C. Hu, L. Weber-Genzel, P. Röttger, F. Kreuter, D. Hovy, B. Plank, “my answer is C”: First-token probabilities do not match text answers in instruction-tuned language models, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 7407–7416. URL: <https://aclanthology.org/2024.findings-acl.441/>. doi:10.18653/v1/2024.findings-acl.441.
- [31] M. Polignano, P. Basile, G. Semeraro, Advanced natural-based interaction for the italian language: Llamantino-3-anita, 2024. URL: <https://arxiv.org/abs/2405.07101>. arXiv:2405.07101.
- [32] R. Orlando, L. Moroni, P.-L. Huguet Cabot, S. Conia, E. Barba, S. Orlandini, G. Fiameni, R. Navigli, Minerva LLMs: The first family of large language models trained from scratch on Italian data, in: F. Dell’Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024), CEUR Workshop Proceedings, Pisa, Italy, 2024, pp. 707–719. URL: <https://aclanthology.org/2024.clicit-1.77/>.
- [33] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, I. Stoica, Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL: <https://arxiv.org/abs/2306.05685>. arXiv:2306.05685.

A. Prompts used

To ensure continuity with the QA items in the Summary-level split, we used essentially the same prompt used in [20], slightly tweaked, for *Local* and *Local alternative* items. For the Summary-level data split (the original INDAQA), the **summary** is provided as context and the model is asked to produce 20 items. Instead, for the Passage-level data split, we provide a **single passage** and ask to produce 3 QA items for *Local* and 1 for *Local (alternative)* setting (Figures 5 and 6, respectively). For *Entity* QA items, we devise a new prompt (Figure 7).

We report the prompt used for multiple-choice conversion (Figure 8) and the QA refinement step (in Figure 9).

```
User prompt
{text}
Domanda: {question}
{choices_block}
Risposta:
```

Figure 4: Prompt used for inference. In the generative task, the `choices_block` parameter is void.

B. Metadata Extraction

We extract bibliographic metadata for each document in our corpus by querying an LLM, using the Wikipedia pages of the documents to ground the generation process. We do not use Wikipedia metadata directly, as we found that for many pages they were missing or not updated with the information in the text. This appendix describes our metadata extraction pipeline and quality assurance procedures.

System prompt

Sei un esperto di letteratura. Il tuo compito è quello di generare domande e risposte sulla trama di un testo letterario.

User prompt

TESTO: {text}

Scrivi almeno 3 domande diverse relative alla trama del testo. Per ogni domanda, scrivi due possibili risposte, entrambe corrette e complete.

Le domande devono essere chiare e non ambigue. Se il testo è breve, genera almeno 2 domande.

Le risposte devono essere brevi e rispecchiare fedelmente il testo originale. Le risposte possono anche essere quasi identiche.

Segui questo formato senza commentare:

Domanda: <domanda>

Risposta A: <risposta>

Risposta B: <risposta>

Assistant prompt

Domanda:

Figure 5: Prompt used to generate the *Local* QA items.

B.1. Extraction Pipeline

For each document in the corpus, we use the opening paragraph from its corresponding Wikipedia page as the primary information source. This introductory text typically contains the essential bibliographic information in a standardized format. We employ an LLM⁸ to extract three metadata fields through separate, targeted prompts:

- **Publication year:** The year of first publication or composition
- **Author name:** The primary author or creator of the work
- **Title:** The canonical title of the work

Each field is queried independently to maximize extraction accuracy and allow for field-specific error handling. We show an example for extracting the publication year in Figure 10. Similar structured prompts are used for author names, titles, and genres, with appropriate formatting constraints specified for each field type.

B.2. Validation and Quality Control

To ensure metadata quality, we cross-reference the information with Wikipedia data. When structured metadata fields are available directly from Wikipedia, we compare the LLM-extracted values against these canonical sources. Discrepancies trigger manual review. When Wikipedia data are unavailable or ambiguous, we perform manual verification and correction.

C. Additional statistics

We report the list of the authorship distribution in Table 6, divided by data split. In the Summary-level split, a few authors dominate the corpus: *Carlo Goldoni* alone accounts for over one-third of the books (35.2%), followed by *Luigi Pirandello* (23.8%) and *Emilio Salgari* (13.3%). The remaining authors each contribute only a small fraction of the data, highlighting a strong skew toward a handful of prolific writers. This imbalance largely reflects the uneven coverage of authors on Wikipedia, as only documents for which a summary was available were included in the dataset.

⁸We found meta-Llama/Llama-3.1-8B-Instruct capable enough for this simple task.

System prompt

Sei un esperto di letteratura. Il tuo compito è quello di generare domande e risposte sulla trama di un testo letterario.

User prompt

TESTO: {text}

Ecco degli esempi di domande e risposte riguardo questo testo:

{question_block}

Scrivi almeno un'altra domanda (diversa) relativa alla trama del testo. Per ogni domanda, scrivi due possibili risposte, entrambe corrette e complete.

Le domande devono essere chiare e non ambigue. Se il testo è breve, genera comunque almeno una domanda.

Le risposte devono essere brevi e rispecchiare fedelmente il testo originale. Le risposte possono anche essere quasi identiche.

Segui questo formato senza commentare:

Domanda: <domanda>

Risposta A: <risposta>

Risposta B: <risposta>

Assistant prompt

Domanda:

Figure 6: Prompt used to generate the *Local (alternative)* QA items.

In the *Passage-level* split, the distribution is more balanced: the top author, *Enrico Castelnovo*, represents 8.1% of the books, and most other authors contribute between 2% and 7%. This indicates that the passage-level split is more diverse with respect to authorship, and less dominated by a few prolific figures, complementing the summary-level data.

We also show the distribution of publication years for the two data splits in Figure 12. The Summary-level split spans a wide temporal range, with a mean publication year around 1814 and a large standard deviation (206 years), reflecting the broad historical coverage of the original material (strong outliers are listed in Table 7). In contrast, the Passage-level distribution is more homogeneous, with a mean publication year around 1900, a small standard deviation (21 years), and no strong outliers.

Finally, we present how the books are divided into 20 equal-probability bins based on text length quantiles and the chosen book for each bin in Figure 11.

D. Dataset examples

In this Section we present four examples taken from INDAQA2, in order to clarify the structure of the dataset.

E. Regexes

In Table 8, we report the regular expressions used to evaluate models in the multiple-choice setting. The list of regexes was defined by inspecting the outputs of selected models and identifying the most common patterns used to introduce the chosen option.

System prompt

Sei un esperto di letteratura. Dati degli estratti e delle domande e risposte riguardanti un'entità (es., un personaggio letterario, un luogo, etc.), scrivi delle nuove domande e risposte che siano più generali (meno specifiche riguardo a dettagli del libro) ma sempre corrette e coerenti.

User prompt

Linee guida per le domande e risposte:

- Le domande devono essere chiare e pertinenti.
- Le domande e le risposte devono essere formulate in italiano corretto.
- Le domande, più che chiedere dettagli specifici, devono riguardare l'arco narrativo del personaggio/il ruolo dell'entità nella storia.

{context_block}

Task:

Scrivi due/tre nuove domande riguardo {entity}, ognuna con una risposta, che siano più generali ma sempre corrette e coerenti secondo le linee guida.

Segui esattamente questo formato senza commentare:

Domanda: <domanda>

Risposta: <risposta>

Assistant prompt

Domanda:

Figure 7: Prompt used to generate Entity QA items. The model is fed a list of passages and already generated QA items about the entity through context_block.

System prompt

Sei un esperto di letteratura. Dato un estratto, una domanda e le risposte giuste che lo riguardano, genera tre "distrattori" (e.g., risposte sbagliate) plausibili.

User prompt

Libro: {title}

Estratto:

{context}

Domanda: {question}

Risposta A: {references[0]}

Risposta B: {references[1]}

Task: scrivi tre risposte sbagliate ma plausibili che potrebbero confondere un lettore, basandoti sull'estratto fornito. Segui esattamente questo formato senza commentare:

Distrattore X: <distrattore_x>

Assistant prompt

Distrattore 1:

Figure 8: Prompt used to generate Entity Question items starting from passages and existing questions.

System prompt

Sei un esperto di letteratura. Dato un estratto, decidi se la domanda e le risposte che lo riguardano sono appropriate secondo le linee guida.

User prompt

Libro: {title}

Estratto:

{context}

Domanda: {question}

Risposta A: {references[0]}

Risposta B: {references[1]}

Linee guida per le domande e risposte:

- La domanda deve contenere abbastanza riferimenti in modo da non essere ambigua: eventi e personaggi VANNO SPECIFICATI BENE con riferimenti temporali, col nome proprio o con caratteristiche descritte nel testo, **in modo da renderla chiara rispetto all'intero libro** (evita termini generici come 'un uomo', 'una donna', 'uno scontro', 'un incontro', etc.).
- La domanda deve essere formulata in modo da richiedere una risposta specifica e non generica.
- La domanda non deve contenere frasi come 'nell'estratto fornito', 'nel libro', 'secondo te', 'cosa pensi di', 'come ti sembra', etc.; deve essere diretta e neutrale.
- Le risposte devono essere ENTRAMBE corrette e complete; possono cambiare solo nel modo in cui sono scritte (parafrasi).
- Le risposte non devono essere contenute nella domanda stessa.
- La domanda e le risposte devono essere formulate in italiano corretto.

Task:

Valuta se la domanda e le risposte sono appropriate secondo le linee guida. Se sono appropriate, rispondi semplicemente con 'OK'. Se non lo sono, fornisci una versione corretta della domanda e delle risposte seguendo le linee guida ed usando l'estratto fornito; riscrivi tutti gli elementi corretti.

Figure 9: Prompt used to correct the generated QA items.

User prompt

Based on the following text from a Wikipedia page, extract only the publication year of the work. Respond with just the year as a four-digit number.

{wikipedia_paragraph}

Assistant prompt

Year:

Figure 10: Prompt used to extract the publication year from the starting paragraph of a Wikipedia page.

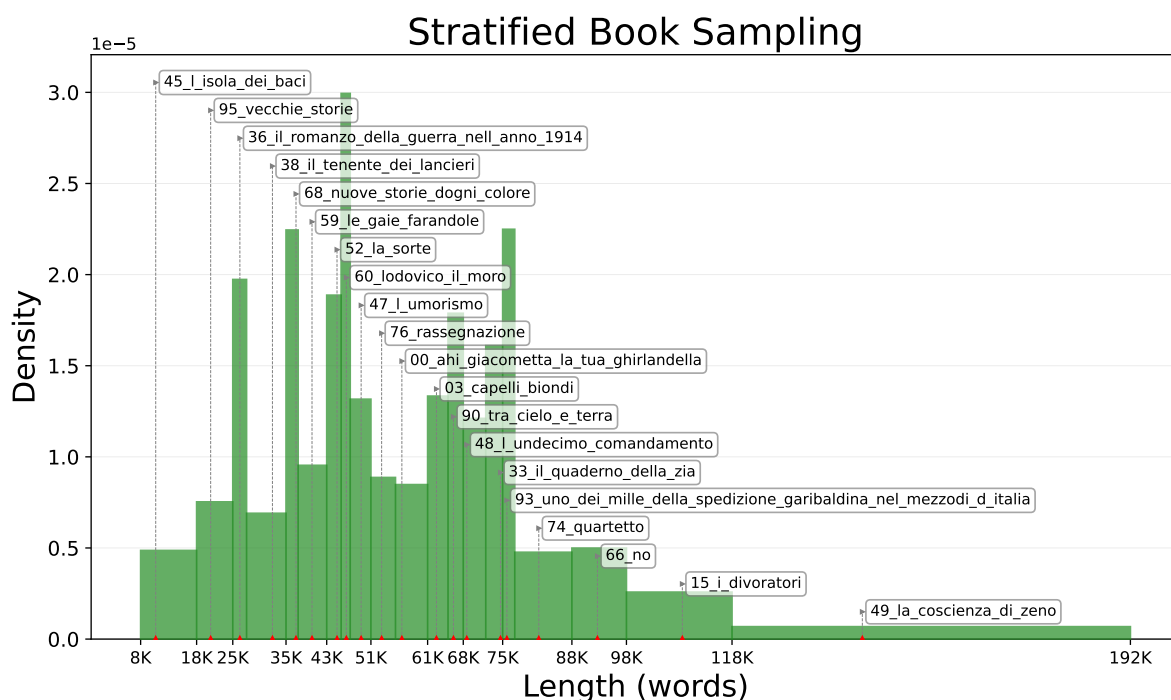


Figure 11: Distribution of the books in the annotation set. We show the ID of the selected 20 books, one from each of the 20 equal-probability bins.

Summary-Level			Passage-Level		
Author	# Books	%	Author	# Books	%
Carlo Goldoni	127	35.2	Enrico Castelnovo	8	8.1
Luigi Pirandello	86	23.8	Matilde Serao	7	7.1
Emilio Salgari	48	13.3	Antonio Beltramelli	5	5.1
Edgar Allan Poe	19	5.3	Edmondo De Amicis	5	5.1
Grazia Deledda	10	2.8	Guido da Verona	5	5.1
Giuseppe Giacosa	7	1.9	Alfredo Panzini	5	5.1
Pietro Metastasio	6	1.7	Luigi Capuana	4	4.0
Jules Verne	4	1.1	Anton Giulio Barrili	4	4.0
William Shakespeare	3	0.8	Alfredo Oriani	3	3.0
Antonio Fogazzaro	3	0.8	Nicola Misasi	3	3.0
Niccolò Machiavelli	2	0.6	Emilio De Marchi	3	3.0
Charles Perrault	2	0.6	Annie Vivanti	3	3.0
Giovanni Verga	2	0.6	F.T. Marinetti	3	3.0
Paolo Mantegazza	2	0.6	Salvatore Farina	2	2.0
Matilde Serao	2	0.6	Lucio D'Ambra	2	2.0
Giovanni Pascoli	2	0.6	Jack London	2	2.0
Alessandro Verri	2	0.6	Gerolamo Rovetta	2	2.0
			Federico De Roberto	2	2.0
			E.A. Butti	2	2.0
			Luigi Pirandello	2	2.0
			Italo Svevo	2	2.0

Table 6

Authors with more than one book in the benchmark, divided by *Summary-level* and *Passage-level* splits, along with the number of books and relative percentages.

Publication year distribution

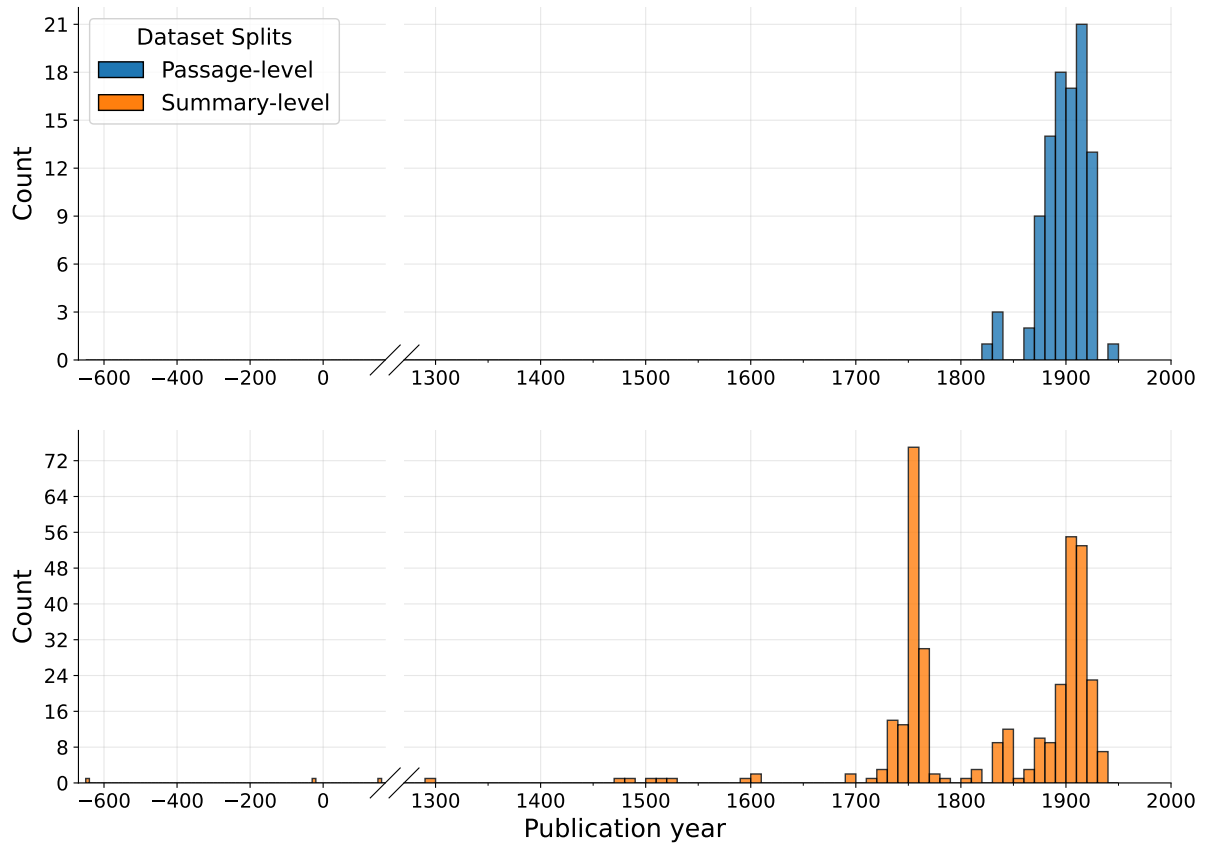


Figure 12: Publication year distribution of the books in the benchmark. Years have been extracted from the introductory paragraph of the related Wikipedia page (either automatically or manually).

Title	Author	Publication year
<i>Lo scudo di Eracle</i>	Esiodo	~650 BCE
<i>Satire</i>	Quinto Orazio Flacco	~30 BCE
<i>La storia vera</i>	Luciano di Samosata	~150 CE
<i>Il Milione</i>	Marco Polo	1298 CE
<i>Stanze per la giostra</i>	Angelo Poliziano	1475 CE
<i>Orlando innamorato</i>	Matteo Maria Boiardo	1483 CE
<i>Arcadia</i>	Jacopo Sannazaro	1504 CE
<i>Mandragola</i>	Niccolò Machiavelli	1518 CE
<i>Clizia</i>	Niccolò Machiavelli	1525 CE

Table 7

Books from the *Summary-level* data split that are outliers with respect to the publication year distribution.

```
// summary_level
{
  "answers": [
    "In un villaggio della Foresta Nera.",
    "Nella Foresta Nera, in un villaggio."
  ],
  "choices": [], // not available
  "entity": null, // not available
  "kind": "summary_question",
  "model": "gemini2-flash",
  "question": "Dove si svolge la festa di fidanzamento iniziale?",
  "question_id": "000_le_villi.summary.0",
  "source_paragraphs_ids": [], // not available
  "source_questions_ids": [], // not available
  "target": {
    "label": null, // not available
    "text": null // not available
  }
}
```

Figure 13: QA item from the summary-level data split. Please note the null/empty data fields for this combination.

```
// passage_level - Local Question item
{
  "answers": [
    "Giacometta Maldì",
    "Giacometta"
  ],
  "choices": [
    "A. Carolina",
    "B. Elena",
    "C. Giacometta Maldì",
    "D. Geltrude"
  ],
  "entity": null, // not available
  "kind": "local_question",
  "model": "gemini-2.5-flash",
  "question": "Come si chiama la giovane donna al centro delle attenzioni per il  
↳ matrimonio?",
  "question_id": "00_ahi_giacometta_la_tua_ghirlandella.set-a.1",
  "source_paragraphs_ids": [0],
  "source_questions_ids": [], // not available
  "target": {
    "label": "C",
    "text": "Giacometta Maldì"
  }
}
```

Figure 14: QA item from the Passage-level data split, Local Question set. Please note the null/empty data fields for this combination.

```
// passage_level - Alternative Local Question item
{
  "answers": [
    "Biondi",
    "Erano biondi"
  ],
  "choices": [
    "A. Neri",
    "B. Biondi",
    "C. Castani",
    "D. Rossi"
  ],
  "entity": null, // not available
  "kind": "local_question_alt",
  "model": "gemini-2.5-flash",
  "question": "Di che colore erano i capelli di Giacometta?",
  "question_id": "00_ahi_giacometta_la_tua_ghirlandella.set-b.1",
  "source_paragraphs_ids": [0],
  "source_questions_ids": [], // not available
  "target": {
    "label": "B",
    "text": "Biondi"
  }
}
```

Figure 15: QA item from the Passage-level data split, Local Alternative Question set. Please note the null/empty data fields for this combination.

```
// passage_level - Entity Questions
{
  "answers": ["La sua eccentricità e la tendenza a comportarsi in modo inappropriato o  
↪ fuori luogo."],
  "choices": [], // not available
  "entity": "adalgisa",
  "kind": "entity_question",
  "model": "gemini-2.5-flash",
  "question": "Qual è una caratteristica distintiva del personaggio di Adalgisa?",
  "question_id": "00_ahi_giacometta_la_tua_ghirlandella.",
  "source_paragraphs_ids": [4, 8],
  "source_questions_ids": [0, 2, 4],
  "target": {
    "label": null, // not available
    "text": null // not available
  }
}
```

Figure 16: QA item from the Passage-level data split, Entity Question set. Please note the null/empty data fields for this combination.

#	Regex
1	<code>r"Risposta \(?([ABCD])\)?"</code>
2	<code>r"risposta corretta è la \(?([ABCD])\)?"</code>
3	<code>r"risposta corretta è: \(?([ABCD])\)?"</code>
4	<code>r"risposta corretta è \(?([ABCD])\)?"</code>
5	<code>r"risposta \(?([ABCD])\)?"</code>
6	<code>r"Risposta: \(?([ABCD])\)?"</code>
7	<code>r"risposta: \(?([ABCD])\)?"</code>
8	<code>r"Risposta è \(?([ABCD])\)?"</code>
9	<code>r"risposta è \(?([ABCD])\)?"</code>
10	<code>r"Risposta è la \(?([ABCD])\)?"</code>
11	<code>r"risposta è la \(?([ABCD])\)?"</code>
12	<code>r"\n\(?([ABCD])\)?\.\s"</code>
13	<code>r"\n\(?([ABCD])\)?\s"</code>
14	<code>r"\(?([ABCD])\)?\.\s?"</code>
15	<code>r"^[ABCD]\.\?\$"</code>

Table 8

Regex patterns used to extract the answer from the models' responses in the MCQA task.

ProverbIT - Easy to complete, hard to choose: A CALAMITA Challenge

Enrico Mensa^{1,*†}, Lorenzo Zane^{2,†}, Calogero J. Scozzaro¹, Matteo Delsanto¹, Tommaso Milani² and Daniele P. Radicioni¹

¹Department of Computer Science, University of Turin, Turin, Italy

²Independent Researcher

Abstract

We present PROVERBIT, a benchmark designed to evaluate the reasoning capabilities of Large Language Models (LLMs) beyond simple pattern matching. While current models demonstrate high proficiency in text generation, their ability to discriminate between plausible but incorrect options remains understudied. PROVERBIT addresses this gap through a challenging multiple-choice task focused on Italian proverbs. In this setting, models are provided with the beginning of a proverb and must select the correct completion from five options. Crucially, four options are always incorrect distractors, making the fifth option, 'None of the others', the only valid answer. This adversarial design forces models to abandon surface-level heuristics and engage in deeper semantic reasoning to actively discard misleading alternatives. To distinguish between a lack of knowledge and a failure in discriminative reasoning, we also introduce a generative completion baseline, where models simply complete the proverb from its initial fragment. The dataset comprises 100 common Italian proverbs, curated and validated by native speakers.

Keywords

Large Language Models, Reasoning, Multi-choice questions, Italian Proverbs

1. Challenge: Introduction and Motivation

The emergence of Large Language Models (LLMs) and Large Reasoning Models (LRMs) has revolutionized the natural language processing landscape across diverse domains [1]. Yet, while these models exhibit remarkable proficiency handling sophisticated linguistic phenomena [2], substantial uncertainty remains regarding their reliability in processing and interpreting culturally embedded linguistic expressions [3], such as proverbs. A proverb is a short, commonly known saying: it expresses a general truth, piece of wisdom, or practical advice, often based on common sense or cultural experience. The understanding of proverbs thus represents a key milestone in language proficiency, and access to the individual components of a proverb allows for the investigation of both lexical access issues and deeper semantic mechanisms.

Since proverbs are high-frequency patterns, standard completion tasks often yield high performance. While we assess this generative baseline, we moved beyond introducing a more complex challenge, evaluating discriminative selection. In this multiple-choice setting, the model must not only recognize the pattern but also evaluate and dismiss plausible alternatives.

In this work we introduce PROVERBIT to the "Challenge the Abilities of LAnguage Models in ITALian" (CALAMITA) initiative [4, 5]. PROVERBIT is a dataset comprising multiple-choice questions centered on Italian proverbs presented at *CLiC-it 2025* [6, 7]. By manually designing alternative endings for the proverbs, we can systematically categorize error types and patterns. Our findings reveal a significant performance dichotomy: despite demonstrating some familiarity of these proverbs in generative settings, all models exhibit a sharp decline in accuracy when forced to operate within a multiple-choice framework.

EVALITA 2026: 9th Evaluation Campaign of Natural Language Processing and Speech Tools for Italian, Feb 26 – 27, Bari, IT

*Corresponding author.

[†]These authors contributed equally.

✉ enrico.mensa@unito.it (E. Mensa); lorenzozane98@gmail.com (L. Zane); calogerojerik.scozzaro@unito.it (C. J. Scozzaro); matteo.delsanto@unito.it (M. Delsanto); milani.tommaso2004@gmail.com (T. Milani); daniele.radicioni@unito.it (D. P. Radicioni)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Challenge Description

The challenge constitutes mainly a *multiple-choice* task, designed to assess the models' ability to select the correct proverb completion rather than relying on surface-level pattern matching. Specifically, given the beginning of a proverb, models are prompted to choose among several alternative endings that are all syntactically well-formed and, to varying degrees, semantically or stylistically plausible, but never correct. The task thus requires the models to actively evaluate and discard misleading options, identifying 'None of the others' as the only correct choice. This setting shifts the focus from automatic generation to discriminative reasoning, emphasizing fine-grained semantic control and proverb-level understanding.

To contextualize the results of the main task, we also introduce a generative *completion* task as a baseline. In this evaluation, models are asked to directly complete each proverb given its initial fragment. This baseline serves to estimate the models' prior familiarity with the proverbs and to verify whether potential errors in the multiple-choice setting stem from a lack of knowledge rather than from the nature of the selection task itself.

3. Data description

The PROVERBIT dataset¹ is composed of 100 multi-choice questions, each regarding the completion of a specific Italian proverb. To create the dataset, we started from an initial set of 200 common Italian proverbs [8] from which we selected 100 of the most commonly used. This process was carried out by three of the authors, which are all native Italian speakers. Each proverb was then manually split into its *beginning* and its *ending*, with the point of division determined to maintain the proverb's semantic coherence in the initial part while allowing for a clear, unambiguous completion. For each proverb, four distinct incorrect alternative endings were manually created, leveraging the following constructive rationale:

- **A** is an ending that has similar sounds to the original continuation, often with an absurd/nonsensical meaning.
- **B** is a non assonant literal synonym of the original ending.
- **C** is the inverse of the original proverb ending, trying to maintain the assonance when possible.
- **D** is a tautological/trivial ending of the proverb, with no assonance.

For sake of clarity we provide an example in English for each of the aforementioned continuations. Completions for the proverb *Actions speak... louder than words* could be:

- A) prouder than swords
- B) at higher volume compared to speech
- C) quieter than words
- D) when they do

As this example shows, the synonym ending is not built on the figurative meaning of the proverb, but it is the literal synonym of the original ending (e.g., at higher volume compared to speech rather than beyond what words can say). This design was adopted to ensure that models cannot simply rely on surface-level syntactic patterns but must engage in deeper semantic and contextual reasoning to identify the absence of the correct completion.

3.1. Data format

The dataset is organized in a comma-separated values format. Each line contains one *complete proverb*, its *beginning* and *ending* splitted, and the four handcrafted incorrect alternatives.

¹The full dataset can be downloaded at <https://huggingface.co/datasets/emensa/proverbIT>.

3.2. Prompts

Two different prompts were devised for the *completion* and *multi-choice* tasks. For the completion baseline, we adopted a simple prompt that requires the model to directly complete a proverb given its beginning:

Completion Prompt Template (translated)

Complete the proverb exactly:

[*Proverb beginning*]...

Reply with the ending only, do not add further comments.

On the multi-choice prompt we specify that the proverb must be completed *exactly*. Since all provided endings are incorrect, we expect models to always answer E) None of the other answers:

Multi-Choice Prompt Template (translated)

Complete the proverb exactly by choosing from the following options (which have no typing errors) indicating only the letter.

[*Proverb beginning*]...

A) ...[*Assonant ending*]

B) ...[*Synonym ending*]

C) ...[*Inverse ending*]

D) ...[*Trivial ending*]

E) None of the other answers

Do not add comments, the possible answers are only A, B, C, D, E.

For sake of clarity we provide an Italian example [with translation] from the actual dataset.

Example of proverb from the dataset

A buon intenditor,... [To a wise man]

A) ...foche canore [singing seals]

B) ...zero chiacchiere [zero chatter]

C) ...molte parole [many words]

D) ...è chiaro tutto [everything is clear]

E) Nessuna delle altre risposte [None of the other answers]

4. Metrics

For both tasks we employ an accuracy metric which is computed differently depending on the task.

Completion Task. To automatically calculate the accuracy on the *completion* task we compute the edit distance² between the provided completion and the correct ending of the proverb.

Multi-Choice Task. The accuracy of the *multi-choice* task was defined as the ratio of correctly chose options (which is always E) over the multiple choices.

²The implementation from <https://docs.python.org/3/library/difflib.html> was employed.

5. Results, Challenge-Specific Insights and Lessons Learned

The evaluation results are reported in Table 1. Each model was prompted once for each of the 100-samples in PROVERBIT and with a zero shot setting. We adopted a chat template, no system prompt and a temperature of 0.

Model	Completion Results		Multi-Choice Results	
	Accuracy	Correct/Total	Accuracy	Correct/Total
Llama-3.1-70B-Instruct	0.67	67/100	0.14	14/100
Llama-3.1-8B-Instruct	0.20	20/100	0.03	3/100
LLaMAntino-3-ANITA-8B-Inst-DPO-ITA	0.13	13/100	0.05	5/100
Minerva-7B-instruct-v1.0	0.13	13/100	0.00	0/100

Table 1

Results on both the completion and multi-choice tasks.

Performance across models. The performances are overall very low, with Llama 70B being the best performing model (0.14 accuracy). For this bigger model, by comparing the results on the completion tasks vs the multi-choice task, we can observe a sharp drop in performances (from 0.67 to 0.14), indicating that even when the model recognizes and is able to complete a proverb, it is not competent enough to discard the four wrong completions in the multi-choice setting.

While the performance gap between tasks persists for smaller models, their competence is already limited in the completion baseline. A qualitative analysis of the outputs reveals a significant deficiency in instruction following: these models are often unable to adhere to simple constraint of ‘only answer with a letter’. This verbosity is strictly penalized by the edit distance metric, leading to near-zero accuracy scores.

Error Analysis Table 2 gives a detailed report of the answers chosen by each model. The error analysis reveal distinct error patterns and biases among the models. Both Llama 70B and Minerva 7B display a strong tendency to select the ‘Synonym’ distractor (Option B), choosing it 56 and 62 times, respectively. Conversely, Llama 8B exhibits a bias toward the ‘Assonant’ distractor (Option A), selecting it in 43 instances. Additionally, the table highlights a significant instruction-following issue for Minerva 7B, which produced 33 ‘Invalid’ responses, unlike the other models which strictly adhered to the valid letter format.

Model	A	B	C	D	E	Invalid
Llama-3.1-70B-Instruct	10	56	13	7	14	0
Llama-3.1-8B-Instruct	43	20	20	14	3	0
Minerva-7B-instruct-v1.0	5	62	0	0	0	33
LLaMAntino-3-ANITA-8B-Inst-DPO-ITA	31	21	13	30	5	0

Table 2

Absolute number of chosen answers in the PROVERBIT task. A = Assonant, B = Synonym, C = Inverse D = Trivial E = None of the others. E is always the correct answer.

Discussion. The drop in performances between the completion and the multi-choice setting indicates that the higher completion accuracy likely stems from surface-level pattern matching rather than deep semantic comprehension. Since the models fail to filter out wrong completions when presented with distractors, it appears that simple language modeling is insufficient and that more powerful reasoning models are required to solve this task.

6. Limitations

While PROVERBIT serves as a robust benchmark, we also report a few limitations. The dataset is relatively small, consisting of only 100 proverbs. While this size is sufficient for a challenge-oriented benchmark, it could limit the statistical power of the evaluation and may reduce the robustness of the conclusions drawn from the results. The construction of the alternatives in the multi-choice setting was performed manually. While this design choice allows for fine-grained control over the types of alternative endings, it also makes the dataset harder to scale and extend to a larger number of proverbs. Finally, we observe that especially when testing with smaller models, our metrics require precise instruction following, and so string-matching metrics like edit distance inflict severe penalties on models that are unable to suppress conversational filler in favor of the requested output.

7. Ethical issues

Our challenge poses no ethical issues and the PROVERBIT dataset does not contain sensitive topics that could lead to ethical issues such as regional differences or social status.

8. Data license and copyright issues

PROVERBIT dataset is distributed under the Creative Commons License Attribution 4.0 International (CC BY 4.0).

Declaration on Generative AI

During the preparation of this work, the authors used gemini-3-pro-preview in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and takes full responsibility for the publication’s content.

References

- [1] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al., Solving quantitative reasoning problems with language models, *Advances in Neural Information Processing Systems* 35 (2022) 3843–3857.
- [2] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, *ACM transactions on intelligent systems and technology* 15 (2024) 1–45.
- [3] F. D. L. Fornaciari, B. Altuna, I. Gonzalez-Dios, M. Melero, A hard nut to crack: Idiom detection with conversational large language models, in: *Proceedings of the 4th Workshop on Figurative Language Processing (FigLang 2024)*, 2024, pp. 35–44.
- [4] G. Attanasio, P. Basile, F. Borazio, D. Croce, M. Francis, J. Gili, E. Musacchio, M. Nissim, V. Patti, M. Rinaldi, et al., Calamita: Challenge the abilities of language models in italian, in: *Proceedings of the 10th Italian Conference on Computational Linguistics (CLiC-it 2024)*, Pisa, Italy, 2024.
- [5] M. Nissim, D. Croce, V. Patti, P. Basile, G. Attanasio, E. Musacchio, M. Rinaldi, F. Borazio, M. Francis, J. Gili, D. Scalena, B. Altuna, E. Azurmendi, V. Basile, L. Bentivogli, A. Bisazza, M. Bolognesi, D. Brunato, T. Caselli, S. Casola, M. Cassese, M. Cettolo, C. Collacciani, L. D. Cosmo, M. P. D. Buono, A. Esuli, J. Etxaniz, C. Ferrando, A. Fidelangeli, S. Frenda, A. Fusco, M. Gaido, A. Galassi, F. Galli, L. Giordano, M. Goffetti, I. Gonzalez-Dios, L. Gregori, G. Grundler, S. Iannaccone, C. Jiang, M. L. Quatra, F. Lagioia, S. M. Lo, M. Madeddu, B. Magnini, R. Manna, F. Mercorio, P. Merlo, A. Muti, V. Nastase, M. Negri, D. Onorati, E. Palmieri, S. Papi, L. Passaro, G. Pensa, A. Piergentili, D. Poterti, G. Puccetti, F. Ranaldi, L. Ranaldi, A. A. Ravelli, M. Rosola, E. S. Ruzzetti, G. Samò, A. Santilli, P. Santin, G. Sarti, G. Sartor, B. Savoldi, A. Serino, A. Seveso, L. Siciliani, P. Torrioni, R. Varvara,

- A. Zaninello, A. Zanollo, F. M. Zanzotto, K. Zeinalipour, A. Zugarini, Challenging the abilities of large language models in italian: a community initiative, 2025. URL: <https://arxiv.org/abs/2512.04759>. arXiv:2512.04759.
- [6] E. Mensa, L. Zane, C. J. Scozzaro, M. Delsanto, T. Milani, D. P. Radicioni, Easy to complete, hard to choose: Investigating LLM performance on the ProverbIT benchmark, in: C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025, pp. 722–734. URL: <https://aclanthology.org/2025.clicit-1.69/>.
- [7] C. Bosco, E. Jezek, M. Polignano, M. Sanguinetti (Eds.), Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025), CEUR Workshop Proceedings, Cagliari, Italy, 2025. URL: <https://aclanthology.org/2025.clicit-1.0/>.
- [8] F. Caramagna, I 200 proverbi italiani più belli e famosi (con significato), 2025. URL: <https://aforisticamente.com/i-200-proverbi-italiani-piu-belli-e-famosi-con-significato/>.

Author Index

- Alcántara, Tania, 206
Apostol, Elena-Simona, 344
Arab, Omar, 587
Atzeni, Stefano, 734
Atzori, Maurizio, 748
Auletta, Nicla, 59
- Bakhtiyarzadeh, Mahdi, 229, 593
Baralis, Elena, 410
Bardelli, Alessandro P., 639
Basile, Pierpaolo, 442, 522
Basili, Roberto, 78, 452, 532, 654
Bayat, Erfan, 495
Berta, Matteo, 477
Borazio, Federico, 78, 654
Bosco, Cristina, 178
Brenna, Sofia, 96
- Cagliero, Luca, 410, 484
Cardoso-Moreno, Marco, 206
Casazza, Simone, 383
Caselli, Tommaso, 734
Cavagnoli, Stefania, 96
Cerquitelli, Tania, 477
Ciaccio, Cristiano, 392
Cignarella, Alessandra Teresa, 138
Ciminelli, Aurora, 366
Cirillo, Nicola, 550
Comandini, Gloria, 96
Corvino, Giulia, 366
Crescenzi, Ivana, 322
Croce, Danilo, 78, 452, 532, 654
Curri, Mattia, 421
Cutugno, Francesco, 1, 5
Çekiç, Tolga, 639
- Damo, Greta, 273
Dang, Nam, 335
Davì, Ernesto, 770
De Luca, Giacomo, 78
Dell’Orletta, Felice, 392
Delsanto, Matteo, 806
Demirtaş, İrem, 639
Di Gravina, Filippo, 421
Di Gregorio, Francesco Damiano, 421
Di Nunzio, Giorgio Maria, 550, 563, 572, 581, 587, 602, 633
Dingeo, Federico, 238
- Dioguardi, Roberto, 770
Draetta, Lia, 178
- Eronen, Juuso, 302
Esuli, Andrea, 27
- Fasulo, Matteo, 196
Fenu, Matteo, 748
Fernández-Castro, Alberte, 782
Ferrando, Chiara, 178
Ferrara, Salvatore, 770
Festa, Donato, 123
Filannino, Michele, 639
Fontana, Emanuele, 421
- Galassi, Andrea, 639
Gallipoli, Giuseppe, 410
García-Vázquez, Omar, 206
Garufi, Giacomo, 782
Gentili, Camilla, 366
Ghahremani, Kourosh, 251
Ghavidel, Fatemeh, 251
Giobergia, Flavio, 53, 131, 433, 467, 696
Gioffré, Luca, 782
Gleaves, Jorge, 294
Greco, Gabriele, 262
Greco, Salvatore, 138
Gregori, Lorenzo, 759
Gualda, Estrella, 178
- Hasnain, Zunaira, 614
HB, Barathi Ganesh, 302, 721
Hong, Son Bui, 709
Hromei, Claudiu D., 532
- Ibrahim, Hadi, 484
- Jiménez, Diana, 206
- Khashechian, Bitá, 563
Kiet, Vo Tuan, 335
Kim, Minseok, 581
Koudounas, Alkis, 467
- La Quatra, Moreno, 138, 467
Labruna, Tiziano, 169, 686
Le Pollotec, Noé Come Jacques, 344
Lolli, Andrea, 495

- Madeddu, Marco, 178
 Magnini, Bernardo, 96
 Manzi, Federica, 352
 Marafatto, Elena, 782
 Marchiori Manerba, Marta, 138
 Mata, Jacinto, 178
 Mata-Vázquez, Jacinto, 284
 Maule, Sofia, 602
 Meenakshi, , 721
 Melendez, Rene, 302
 Mensa, Enrico, 806
 Mercurio, Alessio, 315
 Miaschi, Alessio, 1, 5, 392
 Milani, Tommaso, 806
 Minh Triet, Nguyen, 623
 Mittone, Gianluca, 16
 Mohammadabad, Ali Saleh, 646
 Moreno-Mendieta, Luis, 206
 Moroni, Luca, 782
 Mousavian Anaraki, Seyed Alireza, 654
 Munis, Evren A., 495, 696
 Musacchio, Elio, 522
 Musumeci, Emanuele, 109
 Muti, Arianna, 322

 Napolitano, Davide, 410
 Nardi, Daniele, 109
 Navigli, Roberto, 782
 Nazarmohsenifakori, Mehregan, 646
 Nhan, Tran Phuoc Thanh, 42
 Nissim, Malvina, 392, 734
 Nozza, Debora, 322

 Ocampo, Nicolás Benjamín, 273

 Pachón-Álvarez, Victoria, 284
 Paci, Walter, 759
 Palmero Aprosio, Alessio, 1, 5
 Papadopulos, Eleni, 686
 Pappacoda, Gianmarco, 639
 Pasquini, Daniele, 78
 Patti, Viviana, 178
 Pedrotti, Andrea, 27
 Phu, Duy Dang, 709
 Phu, Huynh Nguyen, 67
 Pirrone, Roberto, 262, 315, 770
 Porcelli, Andrea, 421
 Ptaszynski, Michal, 302, 721
 Puccetti, Giovanni, 27

 R, Jairam, 721
 Radicioni, Daniele P., 806
 Rai, Shahid Iqbal, 654
 Rajabi, Houman, 251
 Rambelli, Giulia, 1, 5
 Ramponi, Alan, 673
 Razmara, Jafar, 229, 593
 Rinaldi, Matteo, 16
 Rodia, Luca M., 696
 Rosso, Paolo, 14, 178
 Ruiz, Guillermo, 294

 Sabato, Mattia, 495
 Saccone, Valentina, 759
 Saha, Priyam, 509
 Salami, Matin, 696
 Santoro, Alessandro, 262
 Sarti, Gabriele, 392, 734
 Savelli, Claudio, 410, 467, 696
 Scaiella, Antonio, 532
 Scala, Simona, 639
 Schiau, Vladimir, 696
 Scozzaro, Calogero Jerik, 16, 806
 Semeraro, Giovanni, 522
 Shcherbakov, Andriy, 452
 Siciliani, Lucia, 1, 5, 522
 Siddique, Fashad Ahmed, 614
 Siragusa, Irene, 262, 315, 770
 Smirnov, Danil, 563
 Son, Bui Hong, 42, 67
 Sorokoletova, Olga E., 109
 Sosto, Mae, 178
 Speranza, Manuela, 96
 Spezia, Salvatore, 770
 Stranisci, Marco Antonio, 1, 5, 16
 Sánchez, Ricardo Muñoz, 138

 Talluto, Giorgio, 315
 Tedeschini, Luca, 196
 Tekanlou, Hadi Bayrami Asl, 229, 593
 Testa, Davide, 96
 Thin, Dang Van, 42, 67, 335, 623, 709
 Tonelli, Sara, 673
 Tonneau, Juliette, 572
 Torres-León, José A., 206
 Torroni, Paolo, 639
 Traina, Federico, 262
 Truică, Ciprian-Octavian, 344

 U, Reshma, 721

Vandervoort, Valentine G. L., 633
Vezzani, Federica, 550
Visconti, Sara, 352
Viviani, Marco, 238, 366
Vázquez-Ramos, Laura, 284

Yassine, Ali, 410, 484
Zaghi, Claudia, 157
Zane, Lorenzo, 806