W-NUT 2026

The 11th Workshop on Natural User-generated Text (W-NUT 2026)

Proceedings of the Workshop

October 28, 2026

©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-470-5

Introduction

The 11th Workshop on Natural User-generated Text (W-NUT) addresses the evolving challenges at the intersection of noisy, user-generated written content and modern Large Language Models (LLMs). As LLMs are increasingly deployed in real-world applications, they must contend with the messy, informal, and diverse nature of human communication found on social media, forums, messaging platforms, and other user-generated sources. Recent research shows that while LLMs demonstrate impressive capabilities across many NLP tasks, their robustness to noisy and unstructured text remains a critical challenge. This workshop bridges traditional NLP research on noisy text with contemporary challenges in building, evaluating, and deploying robust LLMs that can handle the complexities of natural language as it actually appears in the wild. We have received a total of 51 main workshop submissions, including ARR, of which 25 are included in the proceedings. The workshop will be held in hybrid in-person and virtual modes. We have three invited speakers: Hinrich Schütze, Haneul Yoo, and Maria Antoniak, who have generously agreed to share their ongoing research work. We are very thankful to have them in our workshop. We would like to thank the Program Committee members who reviewed the papers, as well as all of the workshop participants for submitting their work.

Organizing Committee

General Chair

JinYeong Bak, Sungkyunkwan University

Program Chair

Hyeju Jang, Indiana University Indianapolis
Rob van der Goot, IT University of Copenhagen
Alan Ramponi, Fondazione Bruno Kessler

Co-Organizers

Weerayut Buaphet, Vidyasirimedhi Institute of Science and Technology
Wei Xu, Georgia Institute of Technology
Alan Ritter, Georgia Institute of Technology

Program Committee

Reviewers

Piyush Agarwal, Khandaker Mamun Ahmed, Onat Dogan Akca, Emily Allaway, Avinash Goutham Aluguvelly, Antonios Anastasopoulos, Rajeel Ahmad Ansari, Maria Antoniak, Noor Mairukh Khan Arnob

Sivaraaman Balakrishnan, Cooper Ballard, Raghvi Baloni, Ankit Bhattacharjee, Krityapriya Bhaumik, Eduardo Blanco

Nathanael Chambers, Xuanjing Chen, Seongwoo Choi, Manuel R. Ciosici, Danilo Croce

Souvik Das, Anay Abhijit Dombe, Jane Arleth dela Cruz

Micha Elsner

Advitya Gemawat

Anna-Carolina Haensch, Marko Haralovic, Muhammad Mubashir Hassan, Monica Hegde, Markus J. Hofmann, YeongJun Hwang

Diana Inkpen

Aditya Jain, Eshaan Jain, Yangfeng Ji, Chao Jiang, Ishan Jindal

Tomoyuki Kajiwara, Hari Kang, HyunJin Kim, Sai Tulasi Kolapudi

Jaehyeok Lee, Xinyu Li, Lucy H. Lin, Danni Liu, Nikola Ljubešić, Elena Lloret

Bolei Ma, Julina Maharjan, Bitan Majumder, Ruslana Margova, Tanmoy Mazumder, Alexander Mehler, Julia Mendelsohn, Yasuhide Miura, Manuel Montes

Günter Neumann, Vincent Ng, Phuong-Anh Nguyen-Le

Naoki Otani

Rajveer Singh Pall, Gianmarco Pappacoda, Suhyeon Park, Yuval Pinter, Alistair Plum, Alba Pérez-Montero

Nelu D. Radpour, Saman Rahbar, Shreyash Rawat, Federico Ruggeri

Iñaki San Vicente, Danae Sanchez Villegas, Albina Sarymsakova, Bogdan Savelyev, Nils Schwager, Mirco Schönfeld, Anirban Sen, Ashfaq Ali Shafin, Vishal Shah, Prakriti Shankar Shetty, Vincent Siddons, Tanvir Ahmed Sijan, Abhai Pratap Singh, Amr Sobhy, Sangmin Song, Richard Sproat, Man-Ling Sung, Fayeq Jeelani Syed

Mariët Theune, Yashdeep Thorat

Sai P Vallurupalli

ChuanHsin Wang, Claire Wang

Sameer Yadav, Linsey C. Yang

Mike Zhang, Ted Zhang, Zihao Zheng, Xiliang Zhu

Keynote Talk

Language Identification in the (Pretraining) Data Processing Pipeline – State of the Art and Open Problems

Hinrich Schütze
University of Munich
2026-10-28 09:30:00 – Room: **Room G15-16**

Abstract: TBA

Bio: Hinrich Schuetze is Professor for Computational Linguistics and co-director of the Center for Information and Language Processing at the University of Munich (LMU Munich). He received his PhD from Stanford in 1995 for his early research on embeddings (ACL Test-of-Time Paper Award). He then worked on natural language processing and information retrieval technology at Xerox PARC, at several Silicon Valley startups and at Google 1995-2004 and 2008/9. Hinrich is a coauthor of Foundations of Statistical Natural Language Processing (with Chris Manning) and Introduction to Information Retrieval (with Chris Manning and Prabhakar Raghavan). He was awarded a European Research Council Advanced Grant in 2017. Hinrich served as president of the Association for Computational Linguistics in 2020. He is a member of Academia Europaea and a fellow of ACL, ELLIS, ELIZA and HessianAI.

Keynote Talk

Enhancing Multilingual LLMs with Code-Switching

Haneul Yoo
New York University
2026-10-28 14:00:00 – Room: **Room G15-16**

Abstract: Despite recent advances, large language models (LLMs) exhibit uneven performance across languages and often struggle with code-switching—the natural alternation between multiple languages. While code-switching is a common and meaningful mode of human communication, LLMs remain brittle to code-switching inputs, treating them as noise. This talk reframes code-switching as a valuable linguistic signal, exploring how it can be leveraged across the LLM lifecycle, from evaluation, to training, to inference. I will begin with output language alignment (OLA), highlighting systematic failures to respond in the expected language in code-switching contexts. I will also introduce code-switching red-teaming (CSRT), which reveals vulnerabilities in safety and multilingual understanding. I will then present code-switching curriculum learning (CSCL), a data-efficient approach to cross-lingual transfer inspired by human second-language learning. Finally, I will introduce code-switching in-context learning (CSICL), which uses gradual transitions from a target language to English to improve multilingual generalization without additional training.

Bio: Haneul Yoo is an Assistant Professor/Faculty Fellow in the Department of Computer Science, Courant Institute School of Mathematics, Computing, and Data Science at New York University. She is also a Research Fellow with the Social Innovation and Special Projects Team in the Division of Policy, Evaluation and Training, Department of Peace Operations at the United Nations. She received her PhD from the School of Computing at KAIST. With a background in machine learning (ML) and natural language processing (NLP), she has worked on (1) developing language models and methods to improve generalization across languages, (2) designing resources and evaluation frameworks for underrepresented languages and cultures, and (3) applying AI in high-stakes, real-world domains for social good. She has published papers in major NLP conferences, including ACL, EMNLP, and NAACL, and has been named a Rising Star at KAIST exploreCSR supported by Google. She has served as a Social Chair at NeurIPS 2026 and co-organized the Multilingual and Equitable Language Technologies (MELT) workshop at COLM 2025, the Women in Machine Learning (WiML) symposium at ICML 2026, and the Generative AI for the World (GenAI4World) workshop at COLM 2026.

Keynote Talk

When Noise Is the Point: What Humanistic Tasks Reveal About Our Data Pipelines

Maria Antoniak
University of Colorado Boulder
2026-10-28 16:00:00 – Room: **Room G15-16**

Abstract: In computational studies of humanistic tasks such as storytelling and poetry, we often encounter datasets whose features frustrate standard NLP pipelines. In this talk, I'll draw on three settings where my collaborators and I have run into this friction. First, I'll discuss user–chatbot conversations like WildChat, where creative requests are long, detailed, and frequently revised, making these logs a rich but also complicated view of how people write with chatbots. Second, I'll describe our development of a fine-grained narrative annotation scheme for very large-scale, heterogeneous web text in pretraining data, where operationalizing a humanities concept is complicated by data diversity. Third, I'll turn to poetry, where the linearization systems used to build training corpora routinely flatten or delete formatting that poets use deliberately. Across these cases, I'll argue that what counts as noise depends on who is reading and for what purpose, and that taking humanistic perspectives seriously can help us build data practices, annotation schemes, and evaluations that better reflect how people actually use language.

Bio: Maria Antoniak is an Assistant Professor of Computer Science at the University of Colorado Boulder where she is also affiliated with the Department of Information Science and a member of the Boulder NLP Group. Previously, she was a Young Investigator at the Allen Institute for AI and a postdoc at the Pioneer Centre for AI at the University of Copenhagen. She completed her PhD in Information Science at Cornell University and her master's degree in Computational Linguistics at the University of Washington. She has also spent time at places like ETH Zürich, Microsoft Research FATE, Twitter Cortex, and Facebook Core Data Science. Her research is in natural language processing and cultural analytics.

Table of Contents

A Precision-First Whitespace Restoration Pipeline for Noisy Urdu Tafseer Text
Muhammad Mubashir Hassan 1

Are More Emotion Theories Better? Joint Prompting Improves Emotion Categorization Robustness
Albina Sarymsakova and Roman Klinger 10

Beyond Clean Text: Evaluating Encoder and Decoder Robustness for Bangla Event Detection in Noisy Text
Tanvir Ahmed Sijan, S M Golam Rifat, Nayeemul Islam and Md Musfique Anwar 26

CrisisKD: Five-Stage Knowledge Distillation for Aspect-Level Sentiment and Emotion Analysis in Crisis Discourse
Marko Haralovic, Onat Dogan Akca, Salih Eren Yücetürk, Minsi Li and Mariët Theune 49

Discourse-Structural Noise in LLM-Generated Explanations
Raghvi Baloni and Achim Rettinger 65

Do LLMs Give Consistent Opinions? Evaluating Response Reliability Under Varying Likert-Scale Formulations in Survey-Style MCQA
Martin Kandlinger, Bolei Ma, Anna-Carolina Haensch and Matthias Aßenmacher 72

Evaluating Token Probabilities as Confidence Estimators in Human Behavior Simulations: A Case Study in Luxembourgish Next Like Prediction
Nils Schwager, Simon Münker, Alistair Plum and Achim Rettinger 89

From Errors to Precision: A Taxonomy-Guided Approach for LLM Flood Cause Extraction
Jane Arleth dela Cruz, Iris Hendrickx and Martha Larson 101

GeNERscore: Semantic Alignment Metrics for Generative Named Entity Recognition
Gianmarco Pappacoda, Federico Ruggeri and Paolo Torroni 117

How Noisy is Automatic Annotation? Characterizing Label Noise from Generative Models in Multi-Dimensional Spanish Subjectivity Annotation
Alba Pérez-Montero, Elena Lloret and Paloma Moreda Pozo 141

Increasing the Robustness of the Fine-tuned Multilingual Machine-Generated Text Detectors
Dominik Macko, Robert Moro and Ivan Srba 154

LLMs are Easily Startled: Fragility Testing with Lookalike Character Noise
Peter deJonge and Nathanael Chambers 167

Multi-author style detection with stylistic embeddings
Man-Ling Sung, Chak-Fai Li, Damianos Karakos, Frederick E. Bernardin III and Brian Ulicny 182

NERQual: Evaluating the Robustness of Named Entity Recognition Models to Data Quality Issues
Denis Trautner and Philippe Thomas 195

Not All Duplicates Are Coordination: Generic vs. Non-Generic Duplicate Campaigns in Information Operations
Ashfaq Ali Shafin and Khandaker Mamun Ahmed 210

Personalized Emotion Analysis via Prompting Large Language Models with Writer Information
Kentaro Hanafusa, Daisuke Maekawa and Tomoyuki Kajiwara 217

Predicting Steering Vectors and Adapter Weights for Few-Shot Author-Style Transfer
Leonard Popp, Danni Liu, Supriti Sinhamahapatra and Jan Niehues 227

QueryDiagramEval: Evaluating and Benchmarking LLM Diagram Generation under Noisy Conditions
Ted Zhang, Souvik Das, Kai Ni, Aditya Agrawal, Xin Guo and Denis Kochedykov 244

Robust Dual-Signal Fusion: Hybrid Neuro-Symbolic Gating with Compressed Chain-of-Thought Refinement for Irony Detection in Social Media Texts
Ankit Bhattacharjee and Krityapriya Bhaumik 266

RSPC: A Benchmark for Modeling Stress and Psychiatric Conditions in Digitally Mediated Relationships using Psychiatrist Annotations
Parmitha Vangapandu, Sai Ganesh M, Sathwik Narkedimilli, Msvpj Sathvik, Timothy Liu, Simon See and Johannes C Eichstaedt 286

Spoiler Court: Interpretable Spoiler Detection via Adversarial Multi-Agent LLM Judgement
Sangmin Song, Hari Kang, Suhyeon Park, Seongwoo Choi, Sungkyun Im and Sangoh Lee . . 313

The Curse of Multilinguality in Lexical Normalization
Saman Rahbar 325

Topic Matching in the Wild: Benchmark and Lessons from Real-World ASR Transcripts
Saman Rahbar, Xiliang Zhu, Irvin Cardoza and David Rossouw 332

TurEngMix: A Text Corpus and Benchmark for Turkish-English Code-Mixed Language Identification and Named Entity Recognition
Ilayda Dogan, Phuong-Anh Nguyen-Le and Julia Mendelsohn 341

Where Does Noise Robustness Live? Evidence from Ablation, Role Swaps, and Cross-Lingual Transfer in Code-Mixed Banking NLU
Piyush Agarwal 356

Program

Wednesday, October 28, 2026

09:00 - 09:30 *Opening Remarks*

09:30 - 10:30 *Invited Talk - Hinrich Schütze*

10:30 - 11:00 *Coffee Break & Networking*

11:00 - 12:30 *Presentation - Oral*

12:30 - 14:00 *Lunch and Networking*

14:00 - 15:00 *Invited Talk - Haneul Yoo*

15:00 - 16:00 *Presentation - Poster*

16:00 - 17:00 *Invited Talk - Maria Antoniak*

17:00 - 17:30 *Closing & Next Steps*

A Precision-First Whitespace Restoration Pipeline for Noisy Urdu Tafseer Text

Muhammad Mubashir Hassan
Independent Researcher
Pakistan
mobizaidi123@gmail.com

Abstract

Document conversion and import are a large but under-studied source of real-world noisy text, distinct from the social-media noise usually studied in normalization. We study missing-space artifacts in a low-resource, Arabic-script setting: an Urdu Qur'anic tafseer exported from InPage `.inp` to `.txt` and then to JSON/markdown, in whose output we observe fused-token artifacts—adjacent words appearing as a single whitespace-delimited token (e.g., اورقرآن for اور قرآن, "and the Qur'an"). Restoring spaces is hard: Urdu's cursive Arabic script does not require spaces at word boundaries, and the text interleaves Urdu prose with Qur'anic Arabic, proper names, and markdown, so a careless split can corrupt preservation-sensitive content. We cast this as precision-first normalization and present a confidence-gated pipeline that proposes only space-insertion edits, verifies a character-preservation invariant, and holds protected regions out of automatic correction. On a 1,000-row silver-labeled sample, the high-confidence auto-accept operating point reaches 96.9% strict precision on decisive silver-labeled cases, against 30.5% for an ungated function-word splitter and 39.7% for a frequency-only baseline, with 0 character-altering edits. This suggests that confidence gating is necessary before applying whitespace restoration to preservation-sensitive Urdu–Arabic text. Our labels are model-assisted silver annotations that partly share lexical and corpus-frequency evidence with the pipeline, so we treat the precision figures as provisional rather than as independent estimates, and the derived artifacts can be released for reproducibility.

1 Introduction

Text normalization has been studied largely on user-generated social-media text (Han and Baldwin, 2011; van der Goot et al., 2021), but noise is not only a property of social media. Large volumes of digitized documents—scanned books, legacy desktop-publishing files, and format migrations—introduce their own systematic, real-world noise during conversion and import. We study one such artifact in a low-resource, Arabic-script setting (Darwish et al., 2021): *missing-space* errors in Urdu, where two adjacent words appear fused into a single token in text produced by a pipeline that converts InPage `.inp` desktop-publishing files to plain text and then to a JSON/markdown representation. For example, اورقرآن (*aur Qur'an*, "and the Qur'an") should be اور قرآن, and کاحق (*ka haq*, "the right of") should be کا حق. We observe these fused tokens in the converted representation and attribute them to the InPage-to-text export and the downstream JSON/markdown conversion taken together (§3). Framed for W-NUT, this is multilingual, low-resource lexical normalization on real-world converted text, with the added constraint that the text is *preservation-sensitive*.

Restoring spaces in Urdu is intrinsically hard. Urdu uses a cursive Arabic script in which the visual boundary between words is created by the non-joining behaviour of letters rather than by spaces, so a missing space is often invisible to the eye and cannot be recovered by simple whitespace heuristics. The problem is compounded in tafseer (Qur'anic commentary), where Urdu prose is interleaved with Qur'anic Arabic, hadith quotations, religious terminology, proper names, and markdown formatting. In this mixed Urdu–Arabic setting a careless split can corrupt sensitive or sacred content, so blind correction is unacceptable.

We therefore adopt a *precision-first* stance: it is safer to leave an uncertain spacing error uncorrected than to introduce a wrong split. Our pipeline proposes only space-insertion edits, assigns each a confidence level from corpus-internal lexical evidence, verifies a character-preservation invariant, and holds protected Arabic and markdown regions

Proceedings of the 11th Workshop on Natural User-generated Text, pages 1–9
October 28, 2026 ©2026 Association for Computational Linguistics

out of automatic correction. It *auto-accepts* only high-confidence edits and *defers* or *rejects* the rest.

Contributions.

- We define a preservation-critical whitespace restoration task for mixed Urdu–Arabic tafseer markdown arising from conversion/import artifacts.
- We present a confidence-gated, space-only correction pipeline that prioritizes precision over recall.
- We introduce a character-preservation invariant that verifies no non-whitespace character is modified.
- We evaluate on a 1,000-row silver-labeled sample and show that the high-confidence auto-accept operating point reaches 96.9% precision on decisive silver-labeled cases, while medium/low candidates are unsafe for blind correction; we give an error analysis of why naive function-word splitting fails, and we state explicitly which evidence the silver labels share with the pipeline they evaluate.

2 Related Work

Lexical and noisy-text normalization. Normalizing noisy text has been a central theme of the W-NUT line of research, beginning with lexical normalization of social-media text (Han and Baldwin, 2011) and maturing into multilingual shared tasks such as MultiLexNorm, which normalizes 13 language variants (van der Goot et al., 2021). This work targets user-generated content, where noise stems from informal spelling and abbreviation. We study a complementary source of real-world noise—document conversion and import—which produces systematic, deterministic artifacts (here, fused tokens) rather than stylistic variation.

Urdu word segmentation and space omission. Urdu word segmentation is a long-standing problem because spaces in Urdu are neither necessary nor sufficient to mark word boundaries. Durrani and Hussain (2010) formulate Urdu word segmentation as a joint problem of space insertion and space deletion, and Lehal (2010) specifically targets the space-omission problem in Urdu script. Akram and Hussain (2010) address segmentation as a component of an Urdu OCR system, and Bin Zia et al. (2018) cast Urdu word segmentation as sequence labeling with conditional random fields. Our task overlaps with this line of work but is deliberately narrower and safety-constrained: we only *insert* spaces, we never delete spaces or alter letters, and we operate on mixed Urdu–Arabic markdown where much of the text must be held out from correction entirely.

OCR and post-conversion correction. Correcting artifacts introduced by digitization is closely related to OCR post-correction, surveyed by Nguyen et al. (2021). Rijhwani et al. (2020) show that OCR post-correction is especially valuable in low-resource settings, where models must exploit limited signal and avoid introducing new errors. Our artifacts arise not from OCR but from an InPage-to-text export and downstream JSON/markdown conversion; the correction is nonetheless a post-conversion normalization problem on noisy text, and it shares the low-resource, error-averse character of that setting. A distinguishing feature of our task is that the text is preservation-critical: much of it (Qur'anic Arabic and quoted material) must be excluded from correction rather than normalized.

Positioning. Prior work addresses Urdu segmentation and OCR correction, but our setting differs because correction is constrained to preservation-safe whitespace insertion in mixed Urdu–Arabic markdown. We do not claim a new segmentation model; we contribute a confidence-gated, character-preserving *operating policy* and an evaluation methodology tailored to preservation-critical text.

3 Data and Task

Corpus. The corpus used in this study is an Urdu Qur'anic tafseer known as *Tafsir al Kashaf*, also referred to in some regions and project metadata as *Tafseer Al-Kashif*. The resulting corpus contains 114 documents (one per surah), 28,072 lines, and 1,581,442 whitespace-delimited tokens. Table 1 summarizes the corpus and the candidates our detector produces from it.

From InPage to the study input. The material reached us in three stages. (i) The Urdu publishers and translators supplied the tafseer as InPage `.inp` desktop-publishing files. (ii) These files were exported to plain Unicode text using InPage's built-in text-export function. (iii) The exported text was

Quantity	Value
Documents / surahs	114
Lines	28,072
Tokens	1,581,442
Split candidates	54,139
Protected candidates	16,439
High-confidence candidates	2,134
Medium-confidence candidates	1,572
Low-confidence candidates	50,433
Review sample	1,000

Table 1: Corpus and candidate statistics.

then parsed into a JSON/markdown representation with one document per surah; this converted representation is the input our study operates on. Comparing it against the lexical sequence a reader of the tafseer would expect, we observe fused-token artifacts in the exported and downstream representation: a word pair that should be separated by a space instead appears as a single whitespace-delimited token.

Expected lexical sequence	اور قرآن
Observed converted token	اورقرآن
Proposed restoration	اور قرآن

Because Urdu word boundaries are marked by non-joining letters rather than by spaces, the fused form remains readable and the defect is easy to overlook. We make no claim about how InPage represents or emits spacing internally, and we do not claim that the defect is present in the publishers' original files. The exact stage at which the whitespace is lost cannot be isolated from the converted artifacts available to us, so we conservatively attribute the noise to the InPage-to-text export and the downstream JSON/markdown conversion taken together rather than to the original publisher source. The task studied here is the recovery of word boundaries in the converted representation, not the reverse engineering of InPage itself.

Availability. Redistribution of the full text depends on the source permissions and is out of scope for this paper. To support reproducibility, the derived candidate records, the silver-labeled evaluation sample, aggregate statistics, and the processing scripts can be released independently of the full corpus text.

Task definition. The input is noisy Urdu markdown. The system may perform exactly one kind of edit: *insert spaces*. The following edits are forbidden: letter changes, letter deletion or insertion,

Merged	Translit.	Restored	Gloss
اورقرآن	aur Qur'an	اور قرآن	and the Qur'an
باغوںمیں	baghoṇ meṇ	باغوں میں	in the gardens
کاحق	ka haq	کا حق	the right of
ہیںلیکن	haiṇ lekin	ہیں لیکن	are, but

Table 2: Representative missing-space artifacts with transliteration, the restored (space-inserted) form, and an English gloss.

Arabic normalization, and markdown restructuring. For each suspected merged token the system emits one of three decisions: *auto-accept* (apply the proposed split), *reject* (do not split), or *defer* (hold the token out of automatic correction). Table 2 shows representative merged tokens with transliteration and gloss.

4 Method

Our pipeline has five stages: protected-region detection, candidate detection, confidence assignment, space-only verification, and a decision policy.

Protected-region detection. Because the corpus mixes sacred Arabic with markdown, we first mark regions that must be held out from automatic correction: markdown headings, blockquotes, code fences, tables, links, and images (line level); inline emphasis spans and Qur'anic reference brackets ﴿ … ﴾ (span level); and any token bearing Qur'anic vocalization marks or an end-of-ayah symbol. Ordinary Urdu diacritics that are common in prose (such as the superscript alef in تعالیٰ) are deliberately *not* treated as protection signals, to avoid over-flagging normal text.

Candidate detection. We scan for tokens in which a short Urdu function word (e.g., اور, میں, ہیں, ہے, نے, پر, کے, کی, کا, کو, سے) appears fused to a neighbouring word as a prefix or suffix, and we propose a single space-insertion split.

Corpus-internal lexical evidence. We avoid an external lexicon by using the corpus itself as a frequency lexicon. For a candidate token t that splits into a function word and a remainder r, we use the corpus frequency of t and of r. The seam is chosen to maximize the standalone frequency of r. The guiding intuition is that a genuine missing-space defect is a *rare* fused token whose remainder

Algorithm 1 Per-token correction decision

Require: token t, corpus frequency f, protected flag p
Ensure: decision in {ACCEPT, DEFER, REJECT}
$S \leftarrow$ function-word prefix/suffix seams of t
if $S = \emptyset$ **then return** REJECT ▷ no candidate
end if
$s^* \leftarrow \arg\max_{s \in S} f(\text{remainder}(t, s))$
$y \leftarrow \text{INSERTSPACE}(t, s^*)$
if $\rho(t) \neq \rho(y)$ **then return** REJECT ▷ fails space-only
end if
$c \leftarrow \text{CONFIDENCE}(t, s^*, f, p)$
if $c =$ HIGH **and** rule$(s^*) \notin$ BAD **and** $t \notin$ OVERRIDE **then**
    return ACCEPT y
else if $c =$ MEDIUM **then return** DEFER y
else return REJECT
end if

is a *frequent* standalone word, whereas a frequent fused token is almost certainly a real single word.

Confidence assignment. Let $f(\cdot)$ be the corpus frequency, t the fused token, and r its remainder after removing the matched function word w. We assign confidence with the following thresholds, applied in order: (i) if $w \in \{$وہ, یہ, تو$\}$ (frequent word-initial substrings) or the candidate is in a protected region, confidence is *low*; (ii) if $f(t) \geq 8$, the fused form is an established word, so *low*; (iii) a word-final تھے/تھی/تھا/نے suffix is capped below *high* (the verb-morphology trap, §6); (iv) otherwise *high* if $f(r) \geq 10$ and $f(t) \leq 3$; *medium* if $f(r) \geq 10$ and $f(t) \leq 7$, or $f(r) \geq 3$ and $f(t) \leq 3$; *low* otherwise. Algorithm 1 summarizes the per-token decision.

Space-only verification. Let $\rho(\cdot)$ denote the function that removes all whitespace from a string. Every proposed split y of a token x must satisfy the character-preservation invariant

$$\rho(x) = \rho(y).$$

If the non-whitespace character sequence is unchanged, then no Urdu or Arabic letter was changed, deleted, or added: the edit is space-only. This invariant certifies character preservation only; it does *not* prove that a split is linguistically correct.

Decision policy. The system auto-accepts a split only when it is high confidence, passes space-only verification, lies outside protected regions, does not come from a known-bad rule, is not on an authoritative override list, and has no competing valid seam. Medium-confidence and otherwise uncertain candidates are deferred; low-confidence candidates, known real-word traps, and tokens that would require letter-level correction are rejected.

5 Evaluation

Evaluation sample. We draw a 1,000-row evaluation sample from the 54,139 candidates, stratified across confidence levels, detection rules, protected/unprotected status, and surahs, so that it exercises both safe fixes and dangerous cases rather than only easy examples. All 1,000 rows are labeled (100% coverage).

Labels. Each row receives one label from: `correct_split`, `wrong_split`, `wrong_seam`, `real_word`, `arabic_or_protected`, `proper_name`, `ambiguous`, and `needs_scholar_review`. The first six are *decisive*; `ambiguous` and `needs_scholar_review` denote *deferred* cases that are excluded from precision.

Where the evidence comes from. The silver labels and the pipeline both draw on Urdu lexical and corpus-frequency evidence, so we set out separately what each uses. *(A) Candidate generation (system).* A closed inventory of short Urdu function words is matched as a prefix or suffix of a token, and the seam that maximizes the remainder's standalone corpus frequency is proposed (§4). *(B) Confidence features (system).* The corpus frequencies $f(t)$ and $f(r)$, the protected-region flag, membership of a high-false-positive function-word set, and a word-final verb-morphology suffix test. *(C) Evidence for the silver labels.* The labeling procedure reads the candidate record—token, proposed split, matched function word, $f(t)$, $f(r)$, protected-region flag, and line context—and applies, in order: the authoritative overrides of (D); an inventory of Qur'anic and Islamic proper names; a morphological wrong-seam test based on the enclitic نہیں; the protected-region flag; the high-false-positive function-word set; a frequency test on the fused token $f(t)$; the word-final نے infinitive test; and finally a frequency test on the remainder $f(r)$. It does *not* read the confidence level or the accept/defer/reject decision that it is being used to evaluate. *(D) Contextual audit.* The high-confidence bucket was additionally inspected in context, and the resulting corrections were recorded as authoritative overrides that take precedence over the heuristic label.

Overlap and residual circularity. Table 3 lists which signals are shared. Corpus frequency, the protected-region flag, the high-false-positive function-word set, and the word-final نے heuristic feed both (B) and (C): the silver evaluation is therefore *not* independent of the features it evaluates. Three properties of the procedure limit, but do not remove, the circularity. First, the labeler is blind to the system's confidence level and decision, so no candidate is labeled `correct_split` merely because the detector was confident about it. Second, the labels are multi-class rather than a binarized copy of that decision, and the shared signals enter at different thresholds and answer a different question: the labeler calls a split correct only when the remainder is a clearly common standalone word ($f(r) \geq 25$), against the $f(r) \geq 10$ that the system's high-confidence rule requires. Third, strict precision charges the system for five distinct kinds of error—wrong splits, wrong seams, real words, protected-region tokens, and proper names—so agreement on frequency evidence alone is not sufficient for a row to count as a success.

The residual dependence is directly visible in the high-confidence bucket. Of its 226 decisive rows, all 219 `correct_split` labels were assigned by the remainder-frequency criterion, and all 7 errors were assigned by the contextual audit of (D). The 96.9% figure should therefore be read as saying that a stricter frequency test corroborates the auto-accept decision on 219 of 226 decisive rows, and that contextual inspection identified seven further cases that frequency evidence alone would have endorsed. An error mode shared by the pipeline and the frequency criterion—most plausibly a real Urdu word whose remainder is itself a frequent standalone word—would be scored as a success by both unless the audit caught it. We discuss the consequences in Limitations.

Metrics. We report *strict precision*, the share of decisive rows labeled `correct_split`:

$$\text{strict precision} = \frac{\#\,\texttt{correct\_split}}{\#\,\text{decisive rows}}.$$

Decisive rows include `wrong_split`, `wrong_seam`, `real_word`, `arabic_or_protected`, and `proper_name`, so wrong splits, real-word splits, protected-region splits, and proper-name splits all penalize the system. We report strict precision separately for the high, medium, and low confidence buckets; high-confidence precision is the main auto-accept operating-point metric. We also report the *deferred rate*, (`ambiguous` + `needs_scholar_review`)/total, and the space-only pass rate from the character-preservation check.

Signal	System	Labels
Shared		
Function-word seam inventory	✓	✓
Corpus frequency $f(t)$, $f(r)$	✓	✓
Protected-region flag	✓	✓
High-false-positive function words	✓	✓
Word-final نے morphology	✓	✓
Not shared		
Assigned confidence / decision	✓	—
Proper-name inventory	—	✓
Enclitic seam test (نہیں)	—	✓
In-context audit of high bucket	†	✓

Table 3: Evidence used by the pipeline (*System*) and by the silver-label procedure (*Labels*). The shared rows are the source of the evaluation's dependence on the system (§5); the frequency thresholds differ between the two ($f(r) \geq 10$ vs. $f(r) \geq 25$). †: the audit corrections are also recorded as the authoritative override list consulted by the deployed decision policy (§4); in the evaluation they act on the labels, where they count as errors against the confidence assignment being measured.

6 Results and Error Analysis

Results. Over all 925 decisive rows the blind strict precision is only 30.5% (282/925): a splitter that accepted every candidate regardless of confidence would be wrong most of the time. This baseline is not the operating point. Table 4 breaks precision down by confidence. At the high-confidence auto-accept operating point, the system achieves 96.9% precision on decisive silver-labeled cases, while medium- and low-confidence candidates fall to 36.0% and 0.0%, respectively. The deferred rate is 7.5% (75/1,000). Table 5 gives the full label distribution; the largest decisive class is `real_word` (37.6%), confirming that a naive splitter would most often damage valid Urdu words.

Baselines and ablation. Table 6 compares our confidence gate against simpler policies on the same sample and labels, reporting precision and coverage (the share of the 282 available correct splits captured). An ungated *function-word splitter* that accepts every candidate reaches only 30.5% precision. A *frequency-only* baseline that accepts a split whenever the remainder is a frequent cor-

Confidence	Rows	Decisive	Correct	Precision
High	266	226	219	96.9%
Medium	209	175	63	36.0%
Low	525	524	0	0.0%
All	1,000	925	282	30.5%

Table 4: Strict precision by confidence bucket on the 1,000-row silver-labeled sample. High confidence is the auto-accept operating point; "All" is the blind-splitter baseline.

Label	Count	Share
correct_split	282	28.2%
wrong_split	25	2.5%
wrong_seam	8	0.8%
real_word	376	37.6%
arabic_or_protected	233	23.3%
proper_name	1	0.1%
ambiguous	75	7.5%
needs_scholar_review	0	0.0%

Table 5: Silver label distribution in the 1,000-row review sample.

pus word ($f(r) \geq 10$) improves precision to 39.7% at full coverage; adding a rare-token condition ($f(t) \leq 3$) raises it to 84.1% but still admits protected-region, verb-morphology, and high-false-positive cases. Our full confidence gate reaches 96.9% precision on decisive silver-labeled cases—57.2 points above the frequency-only baseline and 12.8 above the rare-token variant—at the cost of lower coverage. The progression isolates the contribution of each component: frequency evidence, rarity of the fused token, and the protected-region / rule-blacklist filters. Because the target text is preservation-sensitive, we operate at the high-confidence point. All figures are computed against the same silver labels, so the comparison between policies is internally consistent even though the absolute level is not independently verified (§5).

Space-only safety. Table 7 reports the character-preservation check. All 1,000 proposed splits and all 925 decisive splits are character-preserving: 0 character-altering edits, a 100% pass rate. We stress that this certifies preservation only, not linguistic correctness.

Error analysis. The most error-prone rule is `suffix_fw:` نے, with 0% precision on its decisive rows. This is the Urdu *infinitive trap*: a large number of valid Urdu words end in نے, including infinitive and inflected verb forms such as کرنے (*karne*), ہونے (*hone*), دینے (*dene*), رکھنے (*rakhne*), لانے (*lane*), and کہنے (*kahne*). Treating a word-final نے as a seam therefore splits real words. The trap is specific to the attachment side: matched as a prefix, `prefix_fw:` نے reaches 56.0% precision, which is why the confidence rule caps only *word-final* نے below high (§4). The asymmetry is word-specific rather than positional: `prefix_fw:` کی reaches 5.5% against 35.0% for `suffix_fw:` کی, so it is the rule inventory, not the attachment side alone, that carries the signal. The prefix تو (`prefix_fw:` تو) is similarly unsafe because it is a frequent word-initial substring, whereas `prefix_fw:` اور is far more reliable at 81.3% precision because the conjunction اور rarely begins an unrelated single word.

Auditing the high-confidence bucket surfaced frequency-plausible but wrong cases that corpus statistics alone miss: تیراکی (*tairaki*, "swimming") must not be split as تیرا کی; بہکے (*bahke*, "led astray") is a valid word; کامو appears as a cited term/name; آرامکو is a proper/company name (Aramco); and جبرو is a wrong-seam case whose intended read-

Method	Acc.	Corr.	Prec.	Cov.
Function-word splitter (all)	925	282	30.5%	100.0%
Frequency-only, $f(r) \geq 10$	710	282	39.7%	100.0%
+ rare token, $f(t) \leq 3$	271	228	84.1%	80.9%
Confidence-gated (ours)	226	219	**96.9%**	77.7%

Table 6: Baselines and ablation on the 1,000-row silver-labeled sample (decisive rows). *Acc.*: rows accepted; *Corr.*: correct splits; *Prec.*: precision; *Cov.*: share of the 282 available correct splits captured. Each added constraint—a remainder-frequency threshold, a rare-token condition, and finally the full confidence gate (protected-region exclusion, verb-morphology and high-false-positive rule handling)—trades coverage for precision. Our gate reaches 96.9% precision on decisive silver-labeled cases, 57.2 points above the frequency-only baseline.

Check	Items	Failures	Pass rate
Proposed splits	1,000	0	100%
Decisive splits	925	0	100%

Table 7: Space-only preservation checks ($\rho(x) = \rho(y)$, where ρ removes all whitespace).

ing is جبر و (as in *jabr o tafwid*), not جب رو. Two letter-level typo cases were also observed; these are out of scope because the system may only insert spaces, never change letters. These cases share a profile that frequency evidence rewards rather than penalizes: the fused form is rare and the remainder (کی, کے, کو) is among the most frequent words in the corpus, which is exactly the pattern the confidence rule treats as strong evidence of a lost space. They are therefore direct support for the design decision that corpus frequency alone cannot carry the auto-accept decision, and they are the reason the operating point additionally requires the protected-region, rule-blacklist and override filters of §4. Because each of these cases turns on lexical knowledge that no corpus count expresses, they also motivate independent expert-gold annotation in future evaluation.

Towards richer candidate generation. The failures above all originate in the candidate generator, which keys on a small closed inventory of function-word strings and so cannot distinguish a genuinely fused function word from the same letter sequence occurring inside a single word. One signal that is largely orthogonal to both the function-word inventory and the frequency counts is the cursive joining behaviour of Urdu letters: they divide into those that connect to the following letter and those that do not (e.g. ا, د, ر, و), and the latter terminate a cursive cluster wherever they occur. Since Urdu word boundaries are realized visually by these cluster breaks rather than by spaces (§3), the distribution of joining and non-joining letters around a candidate seam carries information about where a lost space plausibly fell: a seam inside a connected cluster is orthographically less plausible than one at a cluster break. Such features are cheap to compute from the characters alone and independent of corpus counts, and we expect them to be a useful additional input for ranking or filtering candidate seams. We did not evaluate this here and leave it to future work.

7 Conclusion

We studied missing-space artifacts observed after an Urdu tafseer was exported from InPage `.inp` to text and converted to JSON/markdown, and we framed their repair as a preservation-critical, whitespace-only normalization task on noisy mixed Urdu–Arabic text. Our evaluation shows that blind correction is unsafe—strict precision over all decisive candidates is only 30.5%—while a confidence-gated policy that auto-accepts only high-confidence, space-only, unprotected edits reaches 96.9% precision on decisive silver-labeled cases. A character-preservation invariant certifies that every proposed edit is space-only, with 0 character-altering edits across all $1{,}000$ proposed splits. Because the silver labels share lexical and corpus-frequency evidence with the pipeline, that figure describes the behaviour of this operating point on this corpus rather than an independent estimate of generalization. The results nonetheless argue for confidence-gated, character-preserving correction rather than unrestricted splitting in sensitive text. Future work should add independent expert-gold validation, enrich candidate generation with orthographic joining features, extend the method to additional corpora and scripts, and study whether the deferred and rejected candidates can be recovered safely without lowering precision.

Limitations

The evaluation labels are model-assisted *silver* annotations, not independent expert-gold labels; although the high-confidence bucket was additionally inspected in context, the medium and low buckets were not independently verified, so their precision figures should be read as provisional. The study uses a single corpus (one Urdu tafseer), and the method, thresholds, and corpus-frequency lexicon are tuned to it; generalization to other Urdu corpora, genres, or scripts is untested. The system deliberately favors precision over recall, so many true missing-space errors are left uncorrected: deferred and rejected candidates are *not* automatically corrected in the current pipeline. The method handles whitespace restoration only; it does not perform spelling correction, OCR letter-error correction, or Arabic normalization, and tokens needing such fixes are out of scope. Finally, redistribution of the full corpus depends on the source permissions; derived annotations, statistics, and scripts can be shared even where the full text cannot.

Dependence of the evaluation on pipeline features. The silver labels were not generated independently of all the features the system uses. As detailed in §5, lexical, morphological, and corpus-frequency evidence informs both the pipeline's confidence assignment and the labeling procedure;

the labeler is blind to the system's confidence level and decision and applies the shared evidence at different thresholds, but the two are correlated rather than independent. In particular, an error mode common to both—a real Urdu word whose remainder happens to be a frequent standalone word—is scored as a success unless the contextual audit catches it, and only the high-confidence bucket was audited. The reported precision may therefore overestimate what an independent expert-gold evaluation would find. We accordingly interpret the 96.9% figure as evidence about the behaviour of the current high-confidence auto-accept operating point on this corpus, not as a fully independent estimate of how the pipeline generalizes. Establishing the latter requires independent expert-gold annotation by qualified Urdu editors, ideally with more than one annotator and an inter-annotator agreement figure; this is the priority for future work, and should precede any application of the pipeline at scale to sensitive text.

Ethical Considerations

The source text was provided in permitted form, in InPage `.inp` format, by the Urdu publishers and translators who prepared it; the spacing artifacts we correct are observed after the InPage-to-text export and the downstream JSON/markdown conversion, and we make no claim about the publishers' original files. The material is religious text, and its handling is preservation-sensitive: our system is explicitly constrained so that it never changes, deletes, or inserts any non-whitespace character, and it holds Qur'anic Arabic and other protected regions out of automatic correction. This design is intended to avoid altering sacred or quoted content. Full-text redistribution should follow the applicable permissions; where the full text cannot be shared, the derived annotations, aggregate statistics, and processing scripts can be released to support reproducibility. We make no claim of religious authority and intend the work purely as noisy-text normalization. Because the current labels are silver rather than expert-gold, and because they are not fully independent of the pipeline they evaluate (§5), we recommend expert review before any corrected text derived from this pipeline is published or distributed.

References

Misbah Akram and Sarmad Hussain. 2010. Word segmentation for Urdu OCR system. In *Proceedings of the Eighth Workshop on Asian Language Resources*, pages 88–94, Beijing, China. Coling 2010 Organizing Committee.

Haris Bin Zia, Agha Ali Raza, and Awais Athar. 2018. Urdu word segmentation using conditional random fields (CRFs). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2562–2569, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, et al. 2021. A panoramic survey of natural language processing in the arab world. *Communications of the ACM*, 64(4):72–81.

Nadir Durrani and Sarmad Hussain. 2010. Urdu word segmentation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 528–536, Los Angeles, California. Association for Computational Linguistics.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 368–378, Portland, Oregon, USA. Association for Computational Linguistics.

Gurpreet Singh Lehal. 2010. A word segmentation system for handling space omission problem in Urdu script. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP)*, pages 43–50, Beijing, China. Coling 2010 Organizing Committee.

Thi Tuyet Hai Nguyen, Adam Jatowt, Mickael Coustaty, and Antoine Doucet. 2021. Survey of post-OCR processing approaches. *ACM Computing Surveys*, 54(6):1–37.

Shruti Rijhwani, Antonios Anastasopoulos, and Graham Neubig. 2020. OCR post correction for endangered language texts. In *Proceedings of the 2020 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), pages 5931–5942. Association for Computational Linguistics.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021. MultiLexNorm: A shared task on multilingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 493–509. Association for Computational Linguistics.

Are More Emotion Theories Better? Joint Prompting Improves Emotion Categorization Robustness

Albina Sarymsakova[1,2] **and Roman Klinger**[2]
[1]Instituto de Estudios Gallegos Padre Sarmiento (IEGPS-CSIC), Santiago de Compostela, Spain
[2]Fundamentals of Natural Language Processing, University of Bamberg, Germany
albina.s@iegps.csic.es, roman.klinger@uni-bamberg.de

Abstract

Emotion analysis systems are expected by users to provide reliable results. Optimizing their robustness, in addition to accuracy, is particularly challenging in noisy environments, such as social media data or other forms of naturally expressed language. Our hypothesis is that robustness can be improved by guiding a language model to pay attention to multiple emotion signals in the input text. Therefore, requesting labels according to multiple emotion theories may improve robustness. To study this hypothesis, we run perturbation experiments on input instances to test the robustness of LLM configurations based on basic emotions only, and jointly with valence–arousal, and cognitive appraisals. We find that adding valence–arousal and cognitive appraisal along with basic emotion instructions to the emotion classification improves robustness, with the effect being most pronounced for adding appraisals. In a further analysis, we find that this improvement is less susceptible to word order perturbations and more to random word substitutions and deletions. These differences hold across multiple LLMs studied in our work.

1 Introduction

Text-based emotion detection connects affective computing with opinion mining, emphasizing the analysis, interpretation, and extraction of emotion cues from subjective text (Canales and Martínez-Barco, 2014; Muhammad et al., 2025). While traditional sentiment analysis is often confined to labeling texts as positive, negative, or neutral (Zhao et al., 2016), text-based emotion detection allows for the recognition of a wider spectrum of emotions (Troiano et al., 2023b; Greschner and Klinger, 2025), by relying on various psychological theories, such as Ekman's discrete model (Ekman and Friesen, 1978; Ekman, 1992), Russell's dimensional model of valence and arousal (Russell, 1980) and appraisal theories (Sherer et al., 1982; Smith and Ellsworth, 1985).

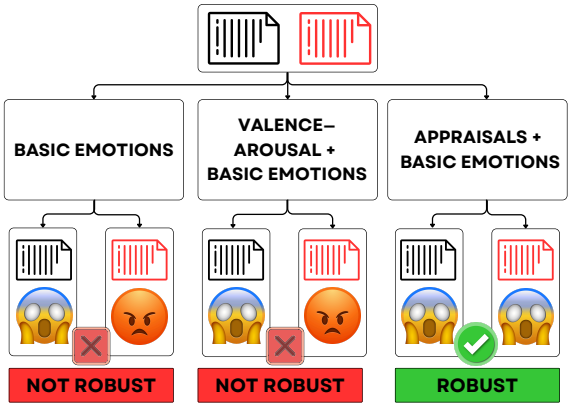


Figure 1: Our goal is to identify which LLM configuration, whether containing basic emotions, valence–arousal + basic, or appraisals + basic, yields more robust emotion-detection outputs between the original text (black) and the perturbed text (red).

In some cases, downstream tasks, such as measuring the potential attention of a car driver through their arousal (Cevher et al., 2019) or interpreting if a piece of art is perceived to be pleasant (Haider et al., 2020) dictate the choice of a psychological theory. Nonetheless, when no concrete downstream task drives the selection, the choice needs to be made based on other signals.

We take the stance that emotion analysis is inherently subjective, and that variances in the output of models or model configurations might appear and be reasonable (often referred to as the phenomenon of human label variation, Plank, 2022). This aspect of affective computing does, however, not change the expectation of users that an automatic system works reliable and produces comparable results across similar texts. Nevertheless, we know that large language model outputs are brittle and sensitive to small meaningless changes to the input texts (as humans are as well, Li et al., 2025a).

Proceedings of the 11th Workshop on Natural User-generated Text, pages 10–25
October 28, 2026 ©2026 Association for Computational Linguistics

We therefore aim at the optimization of the robustness and therefore reliability of emotion classification decisions of prompted large language models. While text-based emotion detection has shown promising results across various textual domains (Acheampong et al., 2021; Al Maruf et al., 2024), including LLMs (Muhammad et al., 2025; Zhang et al., 2025) tests of their robustness (Khurana et al., 2021; Jeon et al., 2025; Sabour et al., 2024; Li et al., 2025a) showed that they exhibit a substantial performance gap relative to humans, particularly due to their sensitivity to noisy texts and contextual variation (Li et al., 2025b). Furthermore, even advanced prompting strategies, such as chain-of-thought approaches, only show limited improvements (Jeon et al., 2025).

The main idea of our work is that prompting large language models according to multiple emotion theories improves their robustness in emotion categorization. The reasoning behind that is that varying emotion theories guide the language model towards different parts of the text to make their classification decision. Theories of affect (Russell, 1980) might, for instance, make the model focus more on adjectives, while appraisal theories (Smith and Ellsworth, 1985) move the focus to events and verbs.

We test this hypothesis by measuring the pair-wise accuracy (Jeon et al., 2025) between predictions on the original and perturbed texts across three LLM-based configurations (basic emotions, valence–arousal+basic, appraisals+basic). We stress that in our work the pair-wise accuracy metric does not quantify correctness: we do not evaluate the labels of perturbed texts against gold emotion labels. Rather, the pair-wise accuracy metric measures how consistent the labels of perturbed texts are with those of the original ones.

This experiment allows us to address the following research questions:

RQ1: Which emotion detection configuration (basic emotions, valence–arousal+basic, appraisals+basic) contributes most to LLMs' predictions robustness under textual perturbation?

RQ2: Which text perturbation method provokes more unrobust labels in these configurations when analyzing their dependency on word, syntax, or broader semantic context changes while predicting emotions?

2 Related Work

2.1 Emotion Analysis and Models

Emotion analysis in natural language processing (NLP) encompasses two major theoretical frameworks: theories of emotion categories and theories of emotion dimensions (Cavicchio, 2025). Each has resulted in numerous available resources for emotion detection (Bostan and Klinger, 2018).

Basic emotion theories conceptualize emotions as discrete categories, defined by how the experiencer displays them. Ekman and Friesen (1978) identified six basic emotions (happiness, fear, sadness, surprise, disgust, and anger) from cross-cultural facial expression studies, which Plutchik (1980) extended to eight primary emotions, adding trust and anticipation, and organized into opposing pairs (e.g., joy vs. sadness, anger vs. fear).

Dimensional theories, in contrast, represent emotions along continuous dimensions that characterize the experiencer's attitude toward an event. Mehrabian and Russell (1974) proposed a three-dimensional model of valence, arousal, and dominance, while Russell (1980) introduced the two-dimensional Circumplex Model of Affect, defined by valence (pleasant–unpleasant) and arousal (calm–activated) alone.

Within the dimensional perspective, cognitive appraisals offer a distinct approach, proposing that emotions arise from subjective assessment of situations (Scherer et al., 2001). It links emotions to appraisal dimensions (Troiano, 2024; Troiano et al., 2023a). Appraisals are dynamic, continuously updated assessments, modifiable at any stage of the emotional response. In contrast to valence–arousal and basic emotion theories, appraisals center on evaluating multiple factors of an event as they relate to the experiencer, such as its suddenness or the experiencer's familiarity with it.

All three theories differ in their perspectives and application to text-based emotion detection. Basic emotion theory addresses this task by tracing emotion lexicons or other explicit lexical cues. The valence–arousal approach rates the experiencer's state along valence and arousal, deriving an emotion from it. In contrast, appraisals evaluate the experiencer–event relationship across a broader set of dimensions (Roseman, 1979; Sherer et al., 1982; Smith and Ellsworth, 1985), demanding the most context reconstruction of the three theories for emotion analysis. This motivated notable progress in appraisal-based approaches for emotion detection

across textual domains, including coping strategies (Troiano et al., 2024), social media (Stranisci et al., 2022), and emotion event self-reports (Hofmann et al., 2020). Troiano et al. (2023b) introduced Crowd-enVent, the most comprehensive appraisal-annotated dataset for event emotion analysis, and Greschner et al. (2026b) developed ContArgA, the largest corpus of arguments annotated for emotions and appraisals.

These studies reflect a growing shift toward the cognitive appraisal approach, which offers a more context- and human-oriented perspective on emotion analysis. This suggests that the appraisal-based setting may yield more robust emotion categorization, since it requires analyzing broader context and properties, part of the hypothesis we explore.

2.2 Robustness Testing

The *robustness* in our study is defined as the evaluation of models' capacity to maintain consistent emotion predictions when the instance's text is perturbed (Pandia and Ettinger, 2021). We emphasize that the notion of robustness in the present study refers to how constant a configuration's prediction on the original and on the perturbed text is, and is therefore orthogonal to correctness against gold emotion labels. A configuration can be robust while consistently reproducing an incorrect label, or accurate on clean text yet volatile under perturbation. We adopt robustness rather than gold-label accuracy as our primary target because our motivating concern is deployment reliability, i.e., whether a system's output changes for noisy or perturbed inputs a user would consider equivalent, a failure mode that persists independently of how accurate a configuration is on non-noisy text, and that has not previously been compared across joint emotion analysis configurations. Existing methods for LLM output robustness testing vary across tasks. Within this context, Liu et al. (2021) explore the language understanding capabilities of LLMs by introducing real-world dialogue perturbations, including linguistic variety, speech characteristics, and noise. For textual perturbations, the authors employed a synonym replacement, random insertion, random swapping, and deletion.

Li et al. (2021) propose a set of perturbation methods, including random substitutions and word insertions, to assess the robustness of pretrained models within the framework of textual adversarial attacks. These perturbation methods are further corroborated by Pandia and Ettinger (2021).

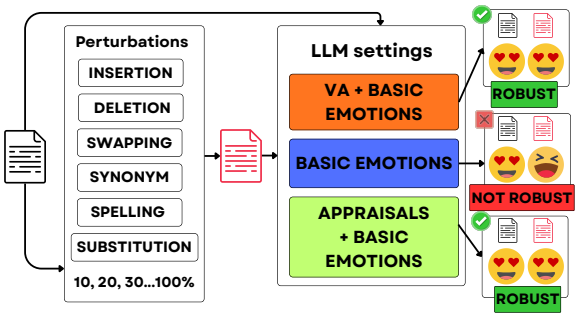


Figure 2: Experimental setup for the robustness study of emotion detection outputs of three configurations: B, VA+B, App+B.

In the context of testing LLMs' sensitivity to emotion context, Jeon et al. (2025) analyze their ability to understand emotions under textual perturbations, such as inserting or substituting words referring to time, place, or agent to create misleading or distracting content. Furthermore, Khurana et al. (2021) evaluated BERT-based models' robustness in sentiment analysis by introducing part-of-speech substitutions, spelling errors (e.g., swapping adjacent characters), and random word swaps. Spelling perturbations are also used for LLM prompt brittleness testing by Li et al. (2025a), across offensiveness and politeness rating, irony detection, and emotion classification tasks.

Based on the aforementioned studies, several strategies are systematically used for robustness testing of LMs, including in emotion analysis to assess sensitivity to textual manipulations and noise. The common techniques include synonym replacement, word swapping, deletion, insertion, substitution, and spelling errors. To the best of our knowledge, these methods have been applied to LMs and LLMs, but none has yet been used to test the robustness of predictions of discrete and dimensional theories that complement one another.

3 Experimental Setup

In the present study, we evaluate the robustness of predictions of three emotion detection configurations. We prompt LLMs for emotion labels under textual perturbations using basic emotions (B), valence–arousal+basic emotions (VA+B), and appraisal+basic emotions (App+B) settings, in order to test their robustness and identify which perturbation method (word-, syntax- or semantic-level) most exposes their sensitivity. The emotion analy-

Configuration	Basic emotions	Appraisals+Basic	VA+Basic
Role & Task	You are an expert in text-based emotion classification. Process the statements below.		
Output format	`emotion`	`22 appraisals, . . . emotion`	`valence, arousal, emotion`
Task description	Single-label classification. Emotion {anger, disgust, fear, guilt, joy, sadness, shame, surprise, trust}.	Task 1: Rate each of 22 appraisals. (1-5 scale). Task 2: *idem* Non-joint.	Task 1: Rate valence & arousal (1-9 scale). Task 2: *idem* Non-joint.
Constraints	Never write "unknown", "neutral", "none", "n/a", or "other". If unsure, pick closest emotion. One line per statement; no blank lines, headers, or explanations. Outputs not conforming to the expected format are marked `INVALID` after 3 retries.		
Example	"I studied all night and still failed the exam." → `sadness`; "Blessed with a baby boy today . . ." → `joy`	"I opened my email and found out that I had been awarded a fully funded research grant I applied for last year." → `4,3,5,5,5,3,1,2,4,4,4,4,3,4,5,4,4,4,2,5,1,1,joy`	"Blessed with a baby boy today . . ." → `7.5,2,joy`; "I can't believe they lied to me." → `2,7,anger`
Input	Now classify: {text1}, {text2}, . . .		

Table 1: Prompt structure.

sis scripts used in this study are publicly available.[1]

Figure 2 provides a visual overview of the experimental workflow. Our experiments utilize the robustness testing strategies detailed in Section 2.2 to perturb selected dataset instances. We perform emotion analysis using LLMs on both the original and perturbed instances. To evaluate the stability of the outputs from both discrete and dimensional models, described in Section 2.1, we employ joint and non-joint prompts to obtain predictions. Finally, we assess robustness by calculating the pair-wise accuracy scores between the emotion labels generated for the original instances versus those for their perturbed counterparts.

3.1 Prompting Method

The prompt structure was adapted from Bagdon et al. (2024) and is summarized in Table 1. The prompts differ across the three emotion detection configurations. Each prompt contains up to six components. The *role and task* establish the model's roles and provide a general instruction on what to do. The *output format* specifies the expected output structure, which varies across configurations: the B outputs only a categorical label; the App+B outputs 22 numerical appraisal ratings (Troiano et al., 2022) followed by a categorical label; and the VA+B outputs integer or decimal ratings for valence and arousal alongside a categorical label. The *task description* provides detailed instructions on how to perform the emotion detection task, and differs across three configurations. The *constraints* define the formatting rules the model must follow and how outputs are parsed: predictions outside the expected format or containing an emotion label outside the predefined set are marked *INVALID* after three unsuccessful retries. Parsing is rule-based: each model's raw response is split into lines, each line is stripped and lower-cased, and, depending on configuration, either matched exactly against the fixed emotion-label set (B) or split on commas and accepted only if it has the expected number of fields with a valid emotion label in the last position (VA+B and App+B); the first line that matches is kept, and the whole request is retried (up to three times) if no line matches. This rule-based parser is identical across all three models. Instance pairs where either the original or perturbed-text label is INVALID are dropped from the pair-wise accuracy computation. Overall dropped percentages per model are: Gemma – 8.05, Llama – 9.45, Qwen – 16.91.[2] The *example* provides labeled in-context instances[3] that illustrate the expected output for each configuration. Finally, the *input* contains the statements the model is asked to process.

We adopted the prompt configuration proposed by Greschner et al. (2026a). It consists of a non-joint setting, in which emotion categories and appraisal dimensions are predicted within a single task, and a joint setting, in which both are predicted together, allowing the two outputs to mutually inform each other. We adopted and adapted this framework, for (i) *Non-joint prompt*, to establish a baseline, using the B configuration emotion detections; and for (ii) *Joint prompt*, to test if VA+B and App+B configuration may benefit from insights from the other emotion theories, while out-

[1] https://www.uni-bamberg.de/en/nlproc/resources/

[2] Because the parsing procedure does not differ across models, we suppose that Qwen shows a higher drop rate in its outputs more often due to the required format under the strict parsing rule described above, most likely on the 23-field App+B task, which is the most constrained of the three output formats (see Table 1).

[3] We use in-context examples since multiple studies (Brown et al., 2020; Bareiß et al., 2024; Creanga et al., 2025; Barfi et al., 2025) highlight that one- and few-shot prompting brings improvements for emotion classification in LLM-based tasks.

Emotion theory	Corpus	Subset
Appraisals	crowd-enVENT	200
	ContArgA	200
Basic emotions	SSEC	200
	ISEAR	200
Valence–Arousal	EmoBank	200
	Facebook VA	200
Total		1200

Table 2: Overview of evaluation corpora, grouped by emotion theory. We select two corpora per theory to test whether results generalize across datasets that share a theoretical grounding but differ in domain and collection method, without implying identical label sets within each group. Each corpus contributes a balanced subset of 200 instances.

put valence–arousal and appraisals rates alongside categorical labels.

3.2 LLMs

We select three LLMs adequate for local deployment for evaluation, based on their established utility and proven performance in text-based emotion classification tasks (Muhammad et al., 2025; Soligo et al., 2026; Yacoubi et al., 2025; Huang et al., 2025; Zhang and Zhong, 2025): Gemma-3-4b (Kamath et al., 2025), Llama-3.3-70b (Grattafiori et al., 2024), and Qwen2.5-7b (Hui et al., 2024). All models are accessed and executed using the ollama (version 0.12.9) library[4]. All models were run at Ollama's default quantization level for each model release (Q4_K_M, 4-bit), without any custom quantization flags.

The decoding configuration was selected to match the model's real-world use, with temperature: 1, top_k: 64, top_p: 0.95. These settings correspond to the default configuration used for the Gemma-3-4b model in the ollama library. We also used the default decoding configuration for Llama-3.3-70b and Qwen2.5-7b with temperature 0.8, top_k 40, and top_p 0.9. The experiments are run on Nvidia A100 GPUs, with a total estimation of 646.35 GPU hours (166.16 h for B, 170.17 h for VA+B, 310.02 h for App+B).

3.3 Emotion Benchmarks and Corpora

This study examines whether B, VA+B, or App+B configuration produces more robust predictions under different textual perturbation methods, and which of these methods acts as more destabilizing at word, syntax, or broader context levels. We select benchmarks and corpora, summarised in Table 2, that are psychologically grounded, empirically well-established, and compatible in terms of emotion categories in order to achieve this goal. We select a subset of corpora that has been used with each theory before.

For the cognitive appraisals benchmark, we use crowd-enVENT (Troiano et al., 2022), which provides ratings across 22 appraisal dimensions (1–5 scale) grounded in componential appraisal theories (Sherer et al., 1982; Smith and Ellsworth, 1985). It covers nine emotion categories (anger, disgust, fear, guilt, joy, sadness, shame, surprise, trust), extending the Ekman set with self-conscious and social emotions closely tied to specific appraisals. We also apply ContArgA (Greschner et al., 2026b), which shares these categories and grounding.

For the basic emotions benchmark, we select SSEC (Schuff et al., 2017) and ISEAR (Scherer and Wallbott, 1994) as the most compatible benchmarks with the appraisal corpora in terms of emotion categories, noting that neither covers the full emotion set: SSEC lacks guilt and shame, while ISEAR omits trust and surprise.

For the valence–arousal benchmark, we use EmoBank (Buechel and Hahn, 2017)[5] and Facebook VA (Preoţiuc-Pietro et al., 2016), restricting the dimensional representation to valence and arousal, the two most consistently identified dimensions of affective studies (Russell, 1980; Posner et al., 2005). Prior work shows that VA scores enable predicting categorical emotion labels with reasonable accuracy (Buechel and Hahn, 2017), which also enables us to evaluate whether the VA-inclusive configuration shows greater robustness in emotion predictions. We use a combination of all corpora for all experiments.

3.4 Perturbation Methods

We use nlpaug library[6] to apply six perturbation methods (see Table 3 for examples): synonym replacement (SR), random swap (RS), random deletion (RD), misspelling augmentation (MA), Conditional Bidirectional Encoder Representations from Transformers random substitution (CBERTRS), and Conditional Bidirectional Encoder Representations from Transformers random insertions (CBER-

[4] https://github.com/ollama/ollama/releases/tag/v0.12.9

[5] While EmoBank uses a 5-point scale, we revert to the original 9-point SAM scale (Bradley and Lang, 1994) in line with standard practice in affective psychology.

[6] https://github.com/makcedward/nlpaug

Method	Text
SR	My biggest is loosing you so delight don river ' thyroxine earn maine ever so present that fearfulness. You cause no absolutely no idea how much you mean to maine.
RS	My biggest so is loosing you don please ' t me ever make face fear. that You no have no absolutely how you idea to much mean me.
RD	My [...] you [...] [...] ' [...] [...] face. [...] have absolutely no how much you to.
MA	Ma biggest is losing you soo please don ' it moke My eve faice that fear. You hvae not absolutely now ideia how much yu meat to ne.
CBERTRS	every secret fear if you only please try ' f cause me ever face that fear. will have an absolutely zero problem how much you enjoy losing me.
CBERTRI	and my biggest is loosing precious you dead so please don ' t again make me have ever to face that fear. may you have my no secrets absolutely scary no fucking idea just how much love you really mean me to please me.

Table 3: Outputs for 50% of the perturbed context for an original instance *My biggest is loosing you so please don't make me ever face that fear. You have no absolutely no idea how much you mean to me*.

TRI). The methods are drawn from various benchmarks, as described below.

Easy Data Augmentation (EDA) (Wei and Zou, 2019) presents SR, which substitutes randomly selected words with synonyms retrieved from a lexical database (Miller, 1994), and it is part-of-speech-aware, meaning a verb is replaced with a verb synonym and a noun with a noun synonym; RS, which randomly selects word positions and swaps each chosen word with its immediate neighbor; and RD, randomly selects words and removes them from the sentence entirely.

MA (Pruthi et al., 2019) replaces randomly selected words with common misspellings from a pre-compiled dictionary of real human spelling errors. The words are chosen randomly, as are the misspellings from the available candidates for that word.

Conditional Bidirectional Encoder Representations from Transformers (CBERT) (Wu et al., 2019; Anaby-Tavor et al., 2019) uses BERT-base-uncased (Devlin et al., 2019) to perform context-aware random substitution (CBERTRS), masking and replacing randomly selected words, and random insertions (CBERTRI), inserting new words at randomly chosen positions between existing tokens. These substitutions and insertions are sampled randomly from the top-k predictions.

3.5 Implementation

We perform experiments to test the robustness of outputs of three emotion detection configurations (B, VA+B, App+B) under textual perturbation. Holding each configuration fixed, we also ablate over the six perturbation methods to determine which type of textual change most destabilizes its predictions to understand whether these configurations are more sensitive to word-level, syntactic, or semantic-level changes.

To perform the experiments, we follow the steps outlined next. We start with an initial dataset of 1.200 instances (see Table 2). We apply textual perturbations using the methods described in Section 3.4. Each of the 1.200 original instances is run 10 times per fraction, yielding 12.000 instances per fraction. Multiplied across 10 incremental fractions of modified content (10%, 20%, ..., 100%), this gives 120.000 instances per perturbation method, and multiplied across 6 perturbation methods, a total of 720.000 perturbed instances. We run emotion predictions on both original and perturbed instances using the prompting methods from Section 3.1 and the LLMs from Section 3.2. We compute pair-wise accuracy between predictions on original and perturbed instances to measure the robustness of the three configurations (B, VA+B, App+B). Finally, we analyze predictions across perturbation methods to evaluate each configuration's sensitivity at three levels: word (SR, MA), syntactic (RS, RD), and semantic (CBERTRS, CBERTRI).

4 Results

In this section, we present the results and analysis following the implementation in Section 3.5, and answer the proposed research questions. We evaluate the robustness of three emotion detection settings (B, VA+B, App+B) by measuring pair-wise accuracy between the original and perturbed texts' emotion outputs. We also analyze which perturbation method, word-level (SR, MA), syntactic-level (RS, RD), or semantic-level (CBERTRS, CBERTRI), most affects robustness, exposing each configurations' sensitivity.

4.1 RQ1: Which emotion detection configuration (basic emotions, valence–arousal + basic, appraisals + basic) contributes most to LLMs' predictions robustness under textual perturbation?

Figure 3 illustrates the average variation in LLM configurations' prediction robustness across perturbation fractions. All three settings (B, VA+B, App+B) show decreasing robustness as the per-

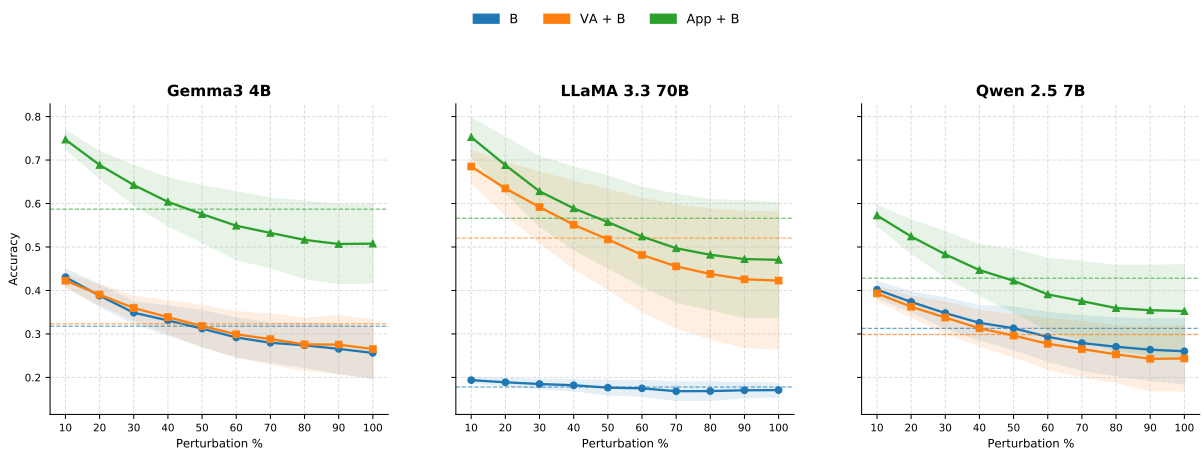


Figure 3: Pair-wise accuracy average curves for emotion predictions between original and perturbed texts. Each panel corresponds to one model (Gemma, LLama, Qwen), with curves comparing three settings (B, VA+B, App+B). Each point is an averaged pair-wise accuracy. For a given perturbation method and perturbation fraction (10%, 20%, 30%, etc.), 10 runs are averaged into one mean per fraction. Repeating this across all 6 methods gives 6 method-level averages per fraction, which are averaged again into the plotted value. The shaded band shows the standard deviation across those 6 method-level averages, not the 10 runs.

turbation fraction increases from 10% to 100%. Nonetheless, notable differences emerge across the three configurations.

The B setting (blue curve), our baseline, yields the lowest accuracy overall, starting below 0.5 for Gemma and Qwen and below 0.2 for Llama at 10% perturbation, and dropping to less than 0.2 at full perturbation. The VA+B setting (orange curve) shows higher accuracy throughout, suggesting that the additional valence–arousal rating task in a joint prompt stabilizes robustness. Within the App+B setting (green curve), all models achieve substantially higher pair-wise accuracy across all perturbation fractions, remaining above roughly 0.3 even at maximum perturbation and approaching 0.8 at 10% fraction for Gemma and Llama. This shows that the appraisal-inclusive configuration produces more robust predictions than either other setting, even as the perturbation fraction increases. It is further validated by the CheckList Invariance test (Ribeiro et al., 2020), specifically by mean fail rates, as Figure 5 shows in Appendix A.

Regarding overall LLMs performance, Llama consistently achieves the highest pair-wise accuracy values across joint configurations and perturbation fractions, while Gemma and Qwen perform comparably, especially for the App+B setting exhibiting the best results. Despite the differences in model sizes, the overall improvement is consistently observed in the App+B setting across all models, suggesting it is more robust even for the smaller models. As for the explanation of the shaded bands representing one standard deviation, the answer to RQ2 provides it.

4.2 RQ2: Which text perturbation method provokes more unrobust labels in these configurations when analyzing their dependency on word, syntax, or broader semantic context changes while predicting emotions?

We ablate across the six perturbation methods to identify which kind of textual change most destabilizes the configuration's (B, VA+B, App+B) predictions, revealing whether a given setting is more sensitive to word-level, syntactic, or semantic changes.

Figure 4 presents the prediction robustness, measured in pair-wise accuracy, of three LLMs across six perturbation types, synonym replacement (SR), random swap (RS), random deletion (RD), misspelling augmentation (MA), and CBERT-based random substitution (CBERTRS) and insertion (CBERTRI), for each emotion detection configuration. Differences in robustness emerge across both perturbation types and configurations. CBERTRS and RD consistently yield the lowest accuracy across all settings, proving the most destabilizing, while RS and SR yield the highest.

The B setting shows the lowest accuracy values overall across all perturbation types. The predictions from the CBERTRS and RD methods in this

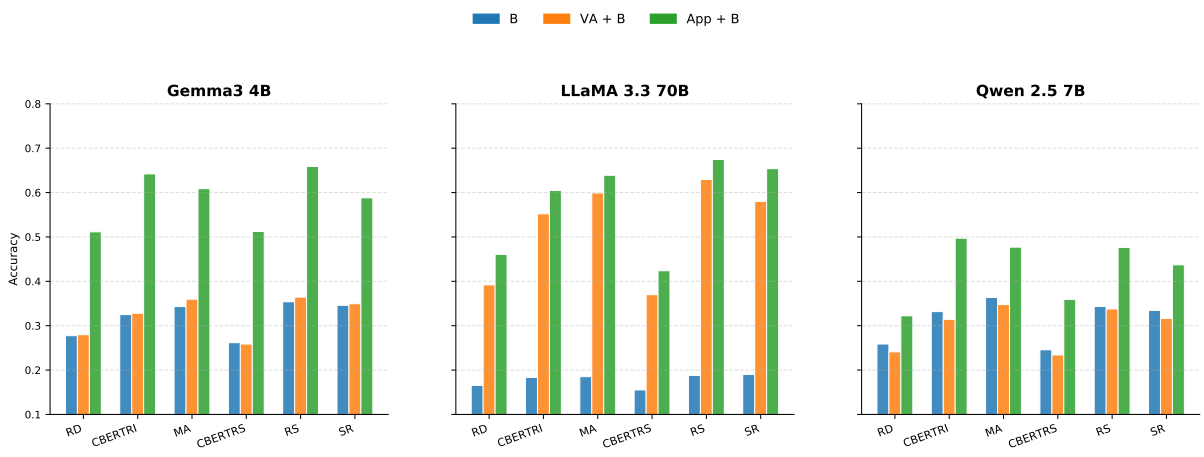


Figure 4: Average pair-wise accuracy by perturbation method and setting. Each bar height is a single average. For one method (e.g., CBERTRI), setting (e.g., VA+B), and model, the data already contains one average pair-wise accuracy per perturbation fraction (10%, 20%, ..., 100%), each itself the average of the 10 runs at that fraction. The bar chart averages those 10 fraction-level values once more into the bar height. A bar thus answers: on average across all fractions from 10% to 100%, how robust were this model's predictions under this perturbation method and setting?

Text	Perturbed	B	B pert.	App+B	App+B pert.
I asked my partner to do a task for me	Asked partner do for me	sadness	trust	**trust**	**trust**
I put someone down without thinking to make myself feel better	put someone without make myself better	surprise	guilt	**guilt**	**guilt**
I failed my driving test that I really wanted to pass	Failed driving I really pass	fear	anger	**sadness**	**sadness**
You'll also find bargain clothes for sale at the markets and on push-carts.	'll bargain clothes markets on-carts.	surprise	trust	**joy**	**surprise**
thinks that for girls boys are easily found. But GIRLS don't realize they need MEN!!	that for girls found. But ' they MEN!!	surprise	joy	**disgust**	**surprise**
Someone vomited on the street.	Vomited on the.	anger	disgust	**disgust**	**disgust**
A mother who shouts at her child for nothing.	A mother for nothing.	disgust	sadness	**anger**	**anger**
Finding out that I passed my final masters exams with a first (distinction)	Finding I final masters (distinction)	anger	sadness	**joy**	**joy**
I felt shame when I ate a lot of food because I was sad rather than hungry.	I felt shame I ate food was sad.	fear	anger	**shame**	**shame**
when i failed my driving test	i failed my driving	fear	sadness	**sadness**	**sadness**

Table 4: Llama predictions for the original and perturbed text of RD of 50% fraction across B and App+B configurations. Labels in bold indicate the predictions of the App+B configuration, which are more robust than those of the B setting.

setting are the least robust. In contrast, RS, CBERTRI, MA and SR perturbations exhibit higher accuracy values, indicating that these perturbation methods enable the B setting to make more robust predictions. The VA+B setting exhibits a similar pattern across perturbation types but maintains slightly higher accuracy values throughout. This configuration also suffers most under the CBERTRS method, followed by the RD, which produce the least robust results.

In the App+B setting, all models achieve substantially higher pair-wise accuracy across all perturbation methods. RS, CBERTRI, MA, and SR remain the least destabilizing, while RD and CBERTRS continue to produce the greatest robustness drops, though considerably smaller than in the other two configurations.

This tendency suggests that the additional context provided in the App+B setting helps the models anchor their predictions more robustly, reducing their sensitivity to surface-level lexical and structural changes. Notably, the most disruptive perturbations are those that alter or remove textual content rather than merely paraphrase it, indicating that the instability stems from genuine shifts in meaning rather than superficial noise. This is further supported by the CheckList Invariance test (Ribeiro et al., 2020), specifically the mean failure rates in Figure 6 (Appendix A).

5 Discussion

Table 4 presents examples and outputs for the B and App+B configurations with Llama, the best-performing model in this study, illustrating how one of the most destabilizing perturbation methods, RD, affects emotion detection. It reveals a consistent pattern for the B setting, our baseline, which proves uniformly unstable across all ten instances. This may stem from the B setting's reliance on specific textual traits that, when disrupted, lose the emotion hints needed for correct identification. Consequently, any perturbation of these elements yields less robust predictions. The B configuration thus appears particularly vulnerable when both syntactic and lexical structure are disturbed, as by random deletion.

The App+B configuration, by contrast, exhibits greater stability, preserving the emotion label in eight of ten cases.[7] Unlike the B setting, which proves to depend more on textual cues, App+B draws on deeper contextual features not strictly tied to lexical-syntactic structure. This broader contextual grounding makes App+B more resilient.

Taken together, these observations indicate that the joint App+B setting owes its greater robustness to drawing on broader contextual cues beyond the lexical-syntactic features the non-joint B setting depends on.

6 Conclusions

We studied whether the robustness of LLM-based emotion detection improves by jointly requesting labels from multiple emotion theories (basic emotions, valence–arousal, or appraisal-based). Our findings show that LLMs are more robust when instructed to predict emotions following more than one theory. Further analysis reveals that the App+B setting considerably stabilizes emotion detection. This effect is most pronounced for word order and least for random word deletions and substitutions.

The divergence between joint (App+B) and non-joint (B) outputs further underscores this effect. While both suffer from random word substitutions and deletions, the non-joint setting relies more on lexical and syntactic cues, yielding less robust predictions, whereas the joint setting captures emotions from a broader contextual basis, yielding more robust outputs. This suggests that combining appraisal and basic emotion theories not only improves robustness but also captures a richer set of emotion cues from the input.

Our research reveals that appraisal and basic emotion theories jointly make LLM predictions more robust. This raises two questions: how LLMs encode appraisals and basic emotions, and which contextual elements this configuration focuses on when predicting. Future research should address both. The first calls for systematic probing of internal model representations. The second demands investigating how the App+B configuration relies on lexical-syntactic context, particularly given that word deletions and substitutions produced the strongest destabilizing effects.

Limitations

Before our emotion-detection configurations can be considered fully robust for text-based emotion analysis, they must be evaluated on a broader scale. Despite promising results in emotion detection robustness, several potential limitations warrant further discussion.

The results we present are based on a limited number of LLMs. Our study is restricted to three of them, Gemma, Llama, and Qwen. Different model families, sizes, and training parameters may respond differently to textual perturbations while predicting emotions, and conclusions drawn from this selection may not be universal. Our model selection also leaves a size gap between the largest model (Llama) and the smaller ones (Gemma and Qwen); adding a mid-size model would help clarify whether the observed robustness gains scale with model size or are specific to the emotion theories tested.

The prompts we used in this study follow a relatively constrained structure, and neither the prompt parameters nor the overall prompt design were systematically varied or optimized. Alternative prompting approaches, such as chain-of-thought or instruction-tuned prompt formats, may yield different robustness patterns for all three configurations.

A further limitation concerns the scope of our robustness assessment. We evaluated the robustness of the categorical emotion labels alone, and did not assess whether the numerical outputs of the VA and APP dimensions within the joint settings are robust in the same manner. This reflects the focus of the present work: our primary interest lies in the categorical emotion labels robustness than in

[7] We also performed a statistical significance test to validate these results, as shown in Appendix B.

the VA and APP dimensional outputs. We therefore leave a systematic evaluation of the robustness of these numerical predictions to future work.

Regarding the text perturbation methods, only six were employed in this study. While these represent a reasonable selection of state-of-the-art approaches, a broader range of perturbation methods could extend and refine the conclusions about which types of textual modifications are most destabilizing across the three configurations.

As noted in Section 2.2, our study evaluates how constant a model's prediction is on the original and on the perturbed text, not correctness against the gold emotion labels that the underlying benchmarks already provide. A configuration could in principle score well on every metric in this paper while being consistently wrong under noisiness. We chose robustness as our primary motivation because it targets deployment reliability rather than clean-text accuracy. An evaluation of robustness alongside gold-label accuracy across configurations is left to future work.

Additionally, all perturbations applied in this study are synthetically generated on selected benchmark texts. Among these benchmarks, there is naturally occurring noisy user-generated text, such as organic typos, slang, or code-switching, since at least one of these benchmarks originates from social media. Nonetheless, we consider evaluating robustness on larger, naturally noisy real-world user-generated corpora an important direction for future work.

Finally, the analysis is constrained to joint and non-joint prompt configurations, namely B, VA+B, App+B. Non-joint configurations, in which basic emotions, valence–arousal, and appraisals are prompted independently, could not be included because there is currently no established method to map their outputs in an all-to-all manner, making direct robustness comparisons across them infeasible.

Ethical Considerations

This work is intended to serve the research community by demonstrating that certain LLM configurations, specifically those in which multiple psychological emotion theories operate in conjunction, can improve the robustness of emotion detection outputs. Our findings aim to illuminate principled criteria for selecting robust configurations for the emotion detection task, thereby reducing unnecessary expenditure of computational resources and time during the configuration search process.

We do not seek to establish the superiority of any emotion theory over the others. Such a claim would be scientifically inappropriate, given that all theories referenced in this work are empirically grounded, theoretically validated, and widely adopted within the research. Rather, our goal is to demonstrate how emotion theories can be mutually reinforcing when applied jointly within LLM-based emotion detection. Similarly, we make no claim that any particular LLM is inherently superior or inferior to others; instead, we examine the extent to which different models may benefit from configurations grounded in combined emotion theories.

Furthermore, emotion analysis carries inherent ethical risks, particularly when applied to manipulate human behavior. These risks may be amplified as LLMs become increasingly capable and convincing in human-computer interaction, potentially enabling more sophisticated forms of emotional manipulation. In our study, we remained mindful of these implications and prioritized transparency and safeguards against misuse when deploying emotion analysis configurations.

All datasets used in this study are publicly available, serve as established benchmarks for emotion analysis in the research community, and contain no personally identifiable information. Their use in this work is strictly limited to advancing research in emotion analysis and does not involve any direct interaction with human participants. All datasets used in this study were employed in accordance with their respective copyright standards, properly cited, and used in compliance with the Creative Commons (CC) license terms.

Finally, AI assistance was used in the preparation of this paper. Specifically, we used the Claude Code extension (version 2.1.143) for Visual Studio Code for programming support, and Claude Sonnet 4.6 for assistance with LaTeX tables formatting and the rephrasing of text for clarity and grammar.

Acknowledgments

This study is funded by CSIC Momentum project (ref. MMT24–IEGPS–01), financed by European Union's Recovery and Resilience Facility-Next Generation, in the framework of the General Invitation of the Spanish Government's public business entity Red.es to participate in talent attraction and retention programmes within Investment 4 of

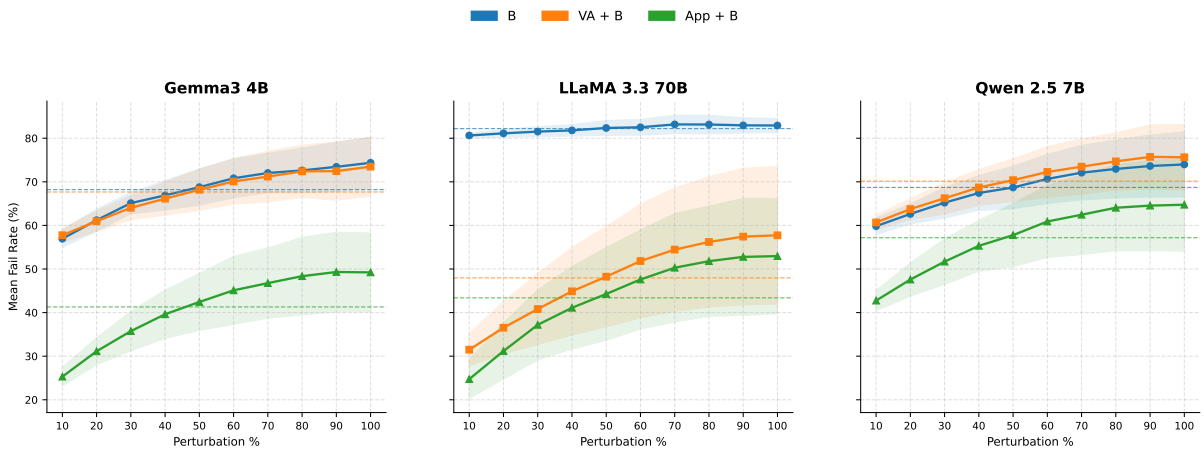


Figure 5: Robustness degradation curves under increasing perturbation fractions per model and setting.

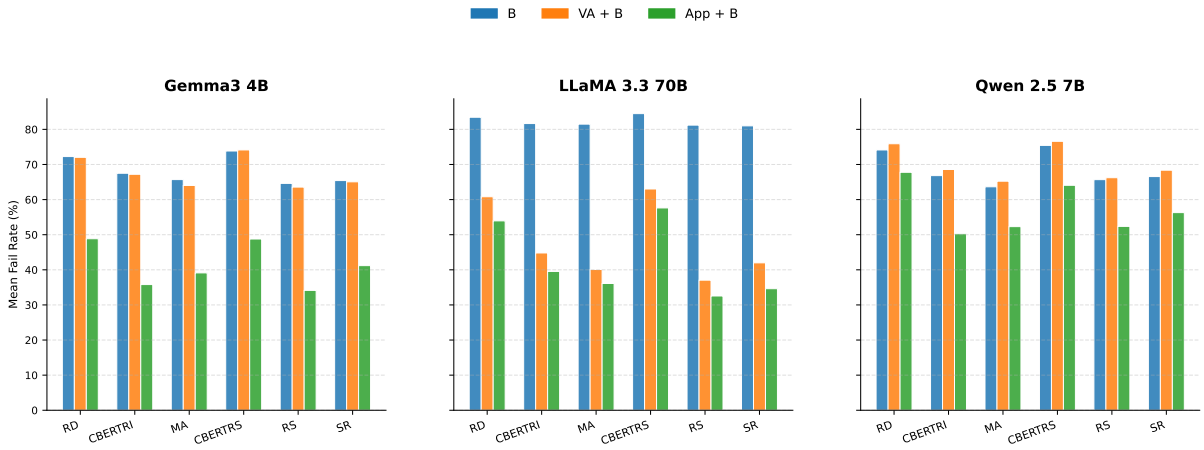


Figure 6: Robustness by perturbation method per model and setting.

Component 19 of the Recovery, Transformation and Resilience Plan. This study was also supported by the "Ministerio de Ciencia, Innovación y Universidades" and "Ayudas para estancias de movilidad en el extranjero José Castillejo para jóvenes doctores 2024" postdoctoral mobility fellowship (ref. CAS24/00327).

A Appendix: CheckList Mean Fail Rate

Figure 5 quantifies emotion predictions robustness using the CheckList framework's Invariance test (Ribeiro et al., 2020), which treats each (original text, perturbed text) pair as a test case and flags a failure whenever the perturbed prediction diverges from the original one. For every model and emotion detection setting (B, VA + B, and App + B), the six perturbation methods (SR, RS, RD, MA, CBERTRS, CBERTRI) were each applied at ten perturbation fractions (10-100%), with ten repeated runs per perturbation fraction. For each (method, fraction) pair, the fail rate was first averaged across the ten runs; these six method-level means were then averaged again within each fraction to produce the plotted point, with the standard deviation across methods rendered as the shaded band. Figure 5 cross-validates that the App + B setting tends to produce more robust emotion labels under perturbations, reported by the pair-wise accuracy metrics in Section 4.1.

Figure 6 summarizes the same INV test results to show which perturbation method degrades each model's label robustness the most. For every (model, emotion detection setting) combination, we obtain a per-run fail rates across all ten fractions and all runs for each method into a single mean. The resulting bars validate previously observed patterns in Section 4.2 that RD and CBERTRS are consistently the most disruptive perturbation methods across all three settings and models. This convergence between two independently computed

Model	Setting	Agree. (A)	Agree. (B)	Diff. (A−B)	Significant
Gemma3	B vs. App+B	0.320	0.583	-0.263	Yes (< 0.0001)
Gemma3	B vs. VA+B	0.320	0.328	-0.007	No (p=0.1614)
Gemma3	App+B vs. VA+B	0.586	0.326	+0.260	Yes (< 0.0001)
Llama	B vs. App+B	0.178	0.552	-0.374	Yes (< 0.0001)
Llama	B vs. VA+B	0.178	0.523	-0.345	Yes (< 0.0001)
Llama	App+B vs. VA+B	0.551	0.522	+0.030	Yes (p=0.0013)
Qwen	B vs. App+B	0.308	0.428	-0.120	Yes (< 0.0001)
Qwen	B vs. VA+B	0.315	0.301	+0.014	Yes (p=0.0343)
Qwen	App+B vs. VA+B	0.428	0.292	+0.137	Yes (< 0.0001)

Table 5: Paired bootstrap significance test resutls. Acc A and Acc B report the pairwise accuracy of the two settings listed in the Setting column: Acc A for the first setting, Acc B for the second. Results with p < 0.05 are treated as significant.

metrics (pair-wise accuracy and Checklist mean fail rates), one measuring prediction robustness (pair-wise accuracy) and the other measuring prediction invariance under perturbation, validates the fragility observed across perturbation methods.

B Appendix: Significance Test Results

We performed a bootstrap test (Koehn, 2004) to make sure our results are statistically significant. Table 5 summarizes these results. For each model, we first compute a per-instance accuracy for each setting (B, App+B, VA+B) between original and perturbed texts, aggregating across all perturbation methods, fractions, and runs. We then compare two settings using a bootstrap test: from the original instances with a valid label and at least one valid prediction in both settings, we draw 10,000 resamples with replacement. The p-value is the fraction of resamples in which the accuracy ranking of the two settings flips relative to the ranking actually observed.

Under this test, the App+B joint setting is significantly more accurate than the B non-joint setting across all three models (Gemma3: $\Delta = +0.263$, $p < 0.0001$; Llama: $\Delta = +0.374$, $p < 0.0001$; Qwen: $\Delta = +0.120$, $p < 0.0001$), indicating that incorporating appraisal theory consistently improves accuracy regardless of perturbation method or underlying model.

References

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, 54(8):5789–5829.

Abdullah Al Maruf, Fahima Khanam, Md Mahmudul Haque, Zakaria Masud Jiyad, Muhammad Firoz Mridha, and Zeyar Aung. 2024. Challenges and opportunities of text-based emotion detection: A survey. *IEEE access*, 12:18416–18450.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, N. Tepper, and Naama Zwerdling. 2019. Do not have enough data? deep learning to the rescue! In *AAAI Conference on Artificial Intelligence*.

Christopher Bagdon, Prathamesh Karmalkar, Harsha Gurulingappa, and Roman Klinger. 2024. "you are an expert annotator": Automatic best–worst-scaling annotations for emotion intensity modeling. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7924–7936, Mexico City, Mexico. Association for Computational Linguistics.

Patrick Bareiß, Roman Klinger, and Jeremy Barnes. 2024. English prompts are better for nli-based zero-shot emotion classification than target-language prompts. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, page 1318–1326, New York, NY, USA. Association for Computing Machinery.

Mobin Barfi, Sajjad Mehrpeyma, and Nasser Mozayani. 2025. UncleLM at SemEval-2025 task 11: RAG-based few-shot learning and fine-tuned encoders for multilingual emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 1563–1569, Vienna, Austria. Association for Computational Linguistics.

Laura Ana Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*,

pages 2104–2119. Association for Computational Linguistics.

Margaret M. Bradley and Peter J. Lang. 1994. Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25 1:49–59.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Sven Buechel and Udo Hahn. 2017. EmoBank: Studying the impact of annotation perspective and representation format on dimensional emotion analysis. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 578–585, Valencia, Spain. Association for Computational Linguistics.

Lea Canales and Patricio Martínez-Barco. 2014. Emotion detection from text: A survey. In *Proceedings of the Workshop on Natural Language Processing in the 5th Information Systems Research Working Days (JISIC)*, pages 37–43, Quito, Ecuador. Association for Computational Linguistics.

Federica Cavicchio. 2025. *Emotion Detection in Natural Language Processing*. Springer.

Deniz Cevher, Sebastian Zepf, and Roman Klinger. 2019. Towards multimodal emotion recognition in german speech events in cars using transfer learning. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019): Long Papers*, pages 79–90, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Claudiu Creanga, Teodor George Marchitan, and Liviu Dinu. 2025. Team Unibuc - NLP at SemEval-2025 task 11: Few-shot text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 468–475, Vienna, Austria. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200.

Paul Ekman and Wallace V Friesen. 1978. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Lynn Greschner, Meike Bauer, Sabine Weber, and Roman Klinger. 2026a. Categorical emotions or appraisals - which emotion model explains argument convincingness better? In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 8190–8203, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Lynn Greschner and Roman Klinger. 2025. Fearful falcons and angry llamas: Emotion category annotations of arguments by humans and LLMs. In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 628–646, Albuquerque, USA. Association for Computational Linguistics.

Lynn Greschner, Sabine Weber, and Roman Klinger. 2026b. Trust me, i can convince you: The contextualized argument appraisal framework and the contarga corpus. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 8327–8346, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Thomas Haider, Steffen Eger, Evgeny Kim, Roman Klinger, and Winfried Menninghaus. 2020. PO-EMO: Conceptualization, annotation, and modeling of aesthetic emotions in German and English poetry. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1652–1663, Marseille, France. European Language Resources Association.

Jan Hofmann, Enrica Troiano, Kai Sassenberg, and Roman Klinger. 2020. Appraisal theories for emotion classification in text. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 125–138, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dawei Huang, Qing Li, Chuan Yan, Zebang Cheng, Zihao Han, Yurong Huang, Xiang Li, Bin Li, Xiaohui Wang, Zheng Lian, and 1 others. 2025. Emotion-qwen: A unified framework for emotion and vision understanding. *arXiv preprint arXiv:2505.06685*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2409.12186*.

Eojin Jeon, Mingyu Lee, Sangyun Kim, Junho Kim, Wanzee Cho, Tae-Eui Kam, and SangKeun Lee. 2025. "going to a trap house" conveys more fear than "going to a mall": Benchmarking emotion context sensitivity for LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14848–14869, Suzhou, China. Association for Computational Linguistics.

Gemma Team Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ram'e, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-Bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gael Liu, and 191 others. 2025. Gemma 3 technical report. *ArXiv*, abs/2503.19786.

Urja Khurana, Eric Nalisnick, and Antske Fokkens. 2021. How emotionally stable is ALBERT? testing robustness with stochastic weight averaging on a sentiment analysis task. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 16–31, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069, Online. Association for Computational Linguistics.

Jiahui Li, Sean Papay, and Roman Klinger. 2025a. Are humans as brittle as large language models? In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2130–2155, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Jiayi Li, Yingfan Zhou, Pranav Narayanan Venkit, Halima Binte Islam, Sneha Arya, Shomir Wilson, and Sarah Rajtmajer. 2025b. Can third parties read our emotions? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 21478–21499, Vienna, Austria. Association for Computational Linguistics.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. Robustness testing of language understanding in task-oriented dialog. In *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)*, pages 2467–2480, Online. Association for Computational Linguistics.

Albert Mehrabian and James A Russell. 1974. *An approach to environmental psychology.* MIT Press.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Shamsuddeen Hassan Muhammad, Nedjma Ousidhoum, Idris Abdulmumin, Seid Muhie Yimam, Jan Philip Wahle, Terry Lima Ruas, Meriem Beloucif, Christine De Kock, Tadesse Destaw Belay, Ibrahim Said Ahmad, Nirmal Surange, Daniela Teodorescu, David Ifeoluwa Adelani, Alham Fikri Aji, Felermino Dario Mario Ali, Vladimir Araujo, Abinew Ali Ayele, Oana Ignat, Alexander Panchenko, and 2 others. 2025. SemEval-2025 task 11: Bridging the gap in text-based emotion detection. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, pages 2558–2569, Vienna, Austria. Association for Computational Linguistics.

Lalchand Pandia and Allyson Ettinger. 2021. Sorting through the noise: Testing robustness of information processing in pre-trained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1583–1596, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In Robert Plutchik and Henry Kellerman, editors, *Theories of Emotion*, pages 3–33. Academic Press.

Jonathan Posner, James A. Russell, and Bradley S. Peterson. 2005. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17:715 – 734.

Daniel Preoţiuc-Pietro, H. Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elisabeth Shulman. 2016. Modelling valence and arousal in Facebook posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 9–15, San Diego, California. Association for Computational Linguistics.

Danish Pruthi, Bhuwan Dhingra, and Zachary C. Lipton. 2019. Combating adversarial misspellings with robust word recognition. In *Proceedings of the 57th*

Annual Meeting of the Association for Computational Linguistics, pages 5582–5591, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Ira J Roseman. 1979. Cognitive aspects of emotion and emotional behavior. In *87th Annual Convention of the American Psychological Association, New York*, volume 58.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Sahand Sabour, Siyang Liu, Zheyuan Zhang, June Liu, Jinfeng Zhou, Alvionna Sunaryo, Tatia Lee, Rada Mihalcea, and Minlie Huang. 2024. EmoBench: Evaluating the emotional intelligence of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5986–6004, Bangkok, Thailand. Association for Computational Linguistics.

Klaus R Scherer, Angela Schorr, and Tom Johnstone. 2001. *Appraisal processes in emotion: Theory, methods, research*. Oxford University Press.

Klaus R. Scherer and Harald G. Wallbott. 1994. Evidence for the Universality and Cultural Variation of Differential Emotion response Patterning. *Journal of Personality and Social Psychology*, 66(2):310–328.

Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Padó, and Roman Klinger. 2017. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 13–23, Copenhagen, Denmark. Association for Computational Linguistics.

Mark Sherer, James E Maddux, Blaise Mercandante, Steven Prentice-Dunn, Beth Jacobs, and Ronald W Rogers. 1982. The self-efficacy scale: Construction and validation. *Psychological reports*, 51(2):663–671.

Craig A Smith and Phoebe C Ellsworth. 1985. Patterns of cognitive appraisal in emotion. *Journal of personality and social psychology*, 48(4):813.

Anna Soligo, Vladimir Mikulik, and William Saunders. 2026. Gemma needs help: Investigating and mitigating emotional instability in llms. *Preprint*, arXiv:2603.10011.

Marco Antonio Stranisci, Simona Frenda, Eleonora Ceccaldi, Valerio Basile, Rossana Damiano, and Viviana Patti. 2022. APPReddit: a corpus of Reddit posts annotated for appraisal. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3809–3818, Marseille, France. European Language Resources Association.

Enrica Troiano. 2024. Where are emotions in text? a human-based and computational investigation of emotion recognition and generation. *Journal for Language Technology and Computational Linguistics*, 37(1):15–26.

Enrica Troiano, Roman Klinger, and Sebastian Padó. 2023a. On the relationship between frames and emotionality in text. *Northern European Journal of Language Technology*, 9.

Enrica Troiano, Sofie Labat, Marco Antonio Stranisci, Rossana Damiano, Viviana Patti, and Roman Klinger. 2024. Dealing with controversy: An emotion and coping strategy corpus based on role playing. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1634–1658, Miami, Florida, USA. Association for Computational Linguistics.

Enrica Troiano, Laura Oberländer, and Roman Klinger. 2023b. Dimensional modeling of emotions in text with appraisal theories: Corpus creation, annotation reliability, and prediction. *Computational Linguistics*, 49(1):1–72.

Enrica Troiano, Laura Oberländer, Maximilian Wegge, and Roman Klinger. 2022. x-enVENT: A corpus of event descriptions with experiencer-specific emotion and appraisal annotations. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1365–1375, Marseille, France. European Language Resources Association.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification 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)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional bert contextual augmentation. In *Computational Science – ICCS 2019*, pages 84–95, Cham. Springer International Publishing.

Ibtissen Yacoubi, Radhia Ferjaoui, Warith Eddine Djeddi, and Anouar Ben Khalifa. 2025. Advancing emotion recognition through llama3 and lora fine-tuning. In *2025 IEEE 22nd International Multi-Conference on Systems, Signals and Devices (SSD)*, pages 348–353.

Jingxiang Zhang and Lujia Zhong. 2025. Decoding emotion in the deep: A systematic study of how llms represent, retain, and express emotion. *arXiv preprint arXiv:2510.04064*.

Yazhou Zhang, Mengyao Wang, Youxi Wu, Prayag Tiwari, Qiuchi Li, Benyou Wang, and Jing Qin. 2025. Dialoguellm: Context and emotion knowledge-tuned large language models for emotion recognition in conversations. *Neural Networks*, 192:107901.

Jun Zhao, Kang Liu, and Liheng Xu. 2016. *Sentiment analysis: Mining opinions, sentiments, and emotions*. MIT Press.

Beyond Clean Text: Evaluating Encoder and Decoder Robustness for Bangla Event Detection in Noisy Text

Tanvir Ahmed Sijan[1,†] **S. M Golam Rifat**[2] **Nayeemul Islam**[3]
Md. Musfique Anwar[1]
[1]Jahangirnagar University, Dhaka, Bangladesh,
[2]Rajshahi University of Engineering & Technology, Rajshahi, Bangladesh
[3]Bangladesh University of Engineering and Technology, Dhaka, Bangladesh,
{sijantanv, golamrifat, nayeemulislam.eee.buet}@gmail.com
manwar@juniv.edu [†]Corresponding author

Abstract

Event detection (ED) systems are typically evaluated on clean, curated text, leaving their robustness to real-world noise largely unexplored, particularly for low-resource languages such as Bangla. We introduce a generalized Bangla news event ontology and a benchmark comprising 9,979 annotated sentences across 40 event subtypes, spanning clean news text, real-world Automatic Speech Recognition (ASR) transcripts, and orthographically corrupted text. We systematically evaluate fine-tuned encoder-only models (BanglaBERT and XLM-R) alongside instruction-tuned decoder-only large language models (Llama 3 and Gemma 3). Our results reveal a clear architectural trade-off: encoder models achieve higher performance on clean text but degrade substantially under noise, whereas decoder-only LLMs are markedly more robust, particularly when event triggers are corrupted. We further show that embedding annotation guidelines during instruction tuning establishes a higher performance baseline on noisy text but yields inconsistent reductions in performance degradation across noisy conditions. Finally, model scaling consistently improves the robustness of decoder-only LLMs, while combined training on clean and noisy data serves as an effective regularization strategy that disproportionately benefits encoder architectures, significantly narrowing the robustness gap.

1 Introduction

The task of event detection (ED) identifies and categorizes events in natural language (Ahn, 2006). It is a fundamental component of information extraction, enabling downstream applications such as emergency monitoring, where timely and accurate extraction of structured events is critical. Despite its importance, most ED research has focused on clean, carefully curated text (Wang et al., 2020; Pouran Ben Veyseh et al., 2022; Yao et al., 2022). In practice, however, textual data is inherently noisy, and even minor typographical errors or ASR transcription mistakes can cause event detection systems to miss critical events (Figure 1).

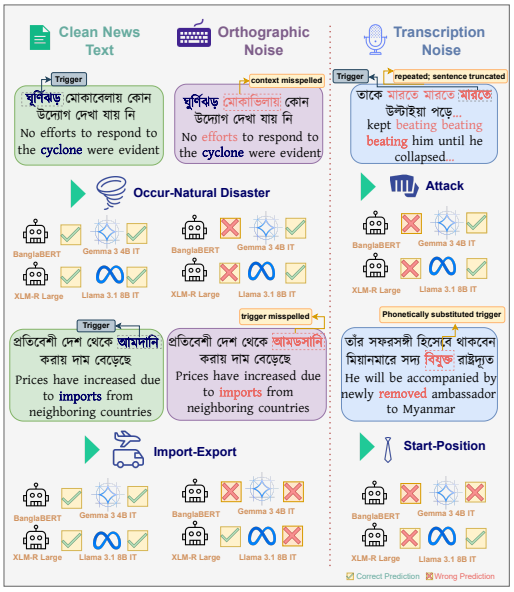


Figure 1: Real-world noise substantially degrades event detection, even when the event trigger itself remains unchanged. Clean examples are drawn from our Clean Test set, while their counterparts are generated using simulated orthographic noise. ASR examples are collected from real Bangla news video transcripts and independently annotated using the same event ontology, introducing naturally occurring transcription artifacts and out-of-distribution vocabulary.

As a low-resource language, Bangla remains underexplored for event detection. Existing studies primarily focus on clean text and narrow domains such as violent incidents (Khandokar et al., 2020; Dey et al., 2021; Ali Khandokar et al., 2025), disasters (Dave et al., 2021), or crime-related events (Hossain et al., 2025). Moreover, there is no generalized Bangla news event ontology comparable to ACE 2005 (Walker, Christopher et al., 2006) for

Proceedings of the 11th Workshop on Natural User-generated Text, pages 26–48
October 28, 2026 ©2026 Association for Computational Linguistics

trigger-based event detection, thereby limiting the systematic study of the task. Research on noisy Bangla text is similarly limited and has largely been restricted to sentence-level sentiment analysis (Islam et al., 2021; Elahi et al., 2024), whereas event detection requires fine-grained token-level identification and classification of event triggers.

Historically, event detection has been dominated by BERT-based encoder models (Wang et al., 2020; Pouran Ben Veyseh et al., 2022; Huang et al., 2024). In contrast, decoder-only LLMs have only recently been adapted to structured extraction through techniques such as code-based representations (Wang et al., 2023), incorporation of sentence-level contextual information (Al Monsur et al., 2026) and instruction tuning with annotation guidelines (Srivastava et al., 2025). Given these advances, an important question emerges regarding how these models compare with their BERT-based counterparts under real-world noisy conditions, particularly for a low-resource language such as Bangla.

To answer this question, we annotate 5,320 sentences from Bangla news articles and 4,659 ASR transcripts from Bangla news videos using same event ontology. We evaluate both encoder-only and decoder-only models for event detection, formulating the latter as a structured code generation task (Wang et al., 2023) with annotation guidelines optionally embedded in the prompt (Sainz et al., 2024; Srivastava et al., 2025). In addition, we simulate orthographic noise using the error generation algorithm of Sifat et al. (2020), which was developed by analyzing common Bangla writing patterns and typographical errors, and evaluate model robustness under varying noise levels.

In summary, our contributions are as follows:

- We develop a generalized Bangla news event ontology and release a human-annotated dataset comprising 9,979 sentences from clean news articles and noisy ASR transcripts, containing 8,041 event mentions across 40 event subtypes.
- We show that although encoder-based models achieve higher performance on clean text, decoder-only LLMs are substantially more robust to noise, particularly when event triggers are corrupted.
- We find that instruction tuning with annotation guidelines consistently improves overall event detection performance but yields limited robustness gains, whereas combined training on clean and ASR data substantially improves encoder robustness and narrows the gap with decoder-only LLMs.

Code and dataset to reproduce our experiments are available.[1]

2 Related Work

Early event detection systems relied on feature engineering and statistical learning techniques (Ahn, 2006; Patwardhan and Riloff, 2009; Hong et al., 2011; Li et al., 2013). More recently, pretrained Transformer encoders such as BERT have become the dominant paradigm, substantially advancing trigger detection performance (Nguyen et al., 2021; Wang et al., 2020; Pouran Ben Veyseh et al., 2022; Huang et al., 2024). Alongside these architectural advances, recent work has also focused on developing event detection datasets spanning diverse domains (Kim et al., 2009; Sims et al., 2019; Le and Nguyen, 2021; Yao et al., 2022) and languages (Pouran Ben Veyseh et al., 2022; Touileb et al., 2024). Large-scale resources such as MAVEN (Wang et al., 2020), RAMS (Ebner et al., 2020), and the TextEE benchmark (Huang et al., 2024) have become standard evaluation platforms for modern event detection research.

Compared with the extensive literature on English event detection, Bangla has received considerably less attention. Existing studies have primarily developed task-specific datasets and models targeting individual application domains, including violent incidents (Khandokar et al., 2020; Dey et al., 2021; Ali Khandokar et al., 2025), disasters (Dave et al., 2021), and crime-related news (Hossain et al., 2025). Moreover, publicly available resources do not provide a generalized news-domain ontology comparable to ACE 2005 (Walker, Christopher et al., 2006), making standardized evaluation across diverse event categories difficult. Studies on noisy Bangla text are likewise limited and have focused primarily on sentence-level tasks such as sentiment analysis (Islam et al., 2021; Elahi et al., 2024) rather than token-level trigger identification.

The rapid progress of decoder-only LLMs has recently motivated their application to event detection. Recent work reformulates the task as

[1] https://github.com/sijantanvir/Bangla-Noisy-Event-Eval

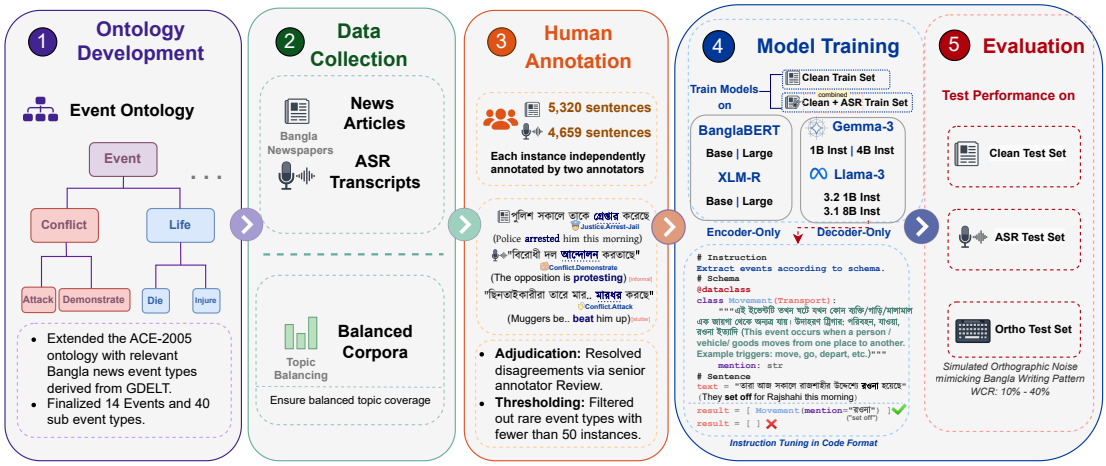


Figure 2: Overview of the dataset construction, training, and evaluation pipeline. We develop a generalized Bangla news event ontology, collect topic-balanced news and ASR corpora, and annotate both using the same event schema. Encoder-only and decoder-only models are trained under two settings: Clean and Combined (Clean + ASR) Training Set. Decoder-only LLMs are instruction-tuned using code-format prompts with and without annotation guidelines embedded as Python docstrings. Robustness is evaluated on the Clean, real-world ASR transcription noise, and simulated orthographic noise test sets.

structured code generation (Wang et al., 2023), explores context-aware encoding (Al Monsur et al., 2026), and employs instruction tuning with annotation guidelines to improve schema understanding and cross-schema generalization (Srivastava et al., 2025). Generative formulations have also been applied to complex event argument extraction (Sharif et al., 2024).

Beyond event extraction, annotation guidelines have also been shown to improve other information extraction tasks, including Named Entity Recognition (Sainz et al., 2024) and Relation Extraction (Pang et al., 2023). However, these methods have almost exclusively been evaluated on clean English benchmarks, leaving the robustness of encoder-only and decoder-only models, as well as the impact of annotation guidelines during instruction tuning, under realistic noisy conditions largely unexplored.

In this work, we address these gaps using Bangla as a representative low-resource language. We introduce a generalized Bangla news event ontology, construct a benchmark comprising clean news articles, real-world ASR transcripts, and simulated orthographic noise, and systematically compare fine-tuned encoder-only and instruction-tuned decoder-only models. We further investigate whether annotation guidelines improve model performance and robustness under noisy conditions.

3 Benchmark Development

3.1 Task Formulation

We evaluate event detection across two distinct modeling paradigms: Token Classification, for encoder-only architectures and Structured Sequence Generation, for decoder-only large language models.

Formally, let an input text sequence of length n be defined as $X = [x_1, x_2, \ldots, x_n]$. Our ontology consists of a predefined set of event types $\mathcal{E}$. The goal of the event detection system is to extract a set of event instances Y from X. Each extracted event instance is defined as a tuple (i, j, t), where $[x_i, \ldots, x_j]$ represents the continuous text span that triggers the event, and $t \in \mathcal{E}$ is the predicted event type. To systematically evaluate the capabilities of different architectures, the extraction of Y is formulated through two paradigms:

Event Detection as Token Classification. The task is framed as a token-level classification problem. A model processes the input text X and assigns a label y_k to each token. The model must simultaneously perform *Trigger Identification* (locating the boundaries i and j of the trigger span) and *Trigger Classification* (mapping the identified span to an event type $t \in \mathcal{E}$).

Event Detection as Structured Sequence Generation. For generative LLMs, the task is reframed as conditional sequence generation. Given the in-

put sentence X, a natural language task instruction I, and the target event schema E_t for a specific event type $t \in \mathcal{E}$, we construct a unified prompt P:

$$P = [I \oplus E_t \oplus X] \quad (1)$$

where $\oplus$ denotes string concatenation. The instruction I dictates the extraction rules and desired structured output format, while E_t provides the semantic definition of the target event. The model is trained to autoregressively generate the structured representation of Y, explicitly handling both the extraction of valid triggers and the rejection of schemas not present in X.

3.2 Ontology Development

We adopted the ACE 2005 (Walker, Christopher et al., 2006) event ontology as the foundation of our schema and extended it with event types relevant to Bangladeshi news but not explicitly covered by ACE (e.g., disasters, festivals, socio-economic events). To identify these event types, we analyzed Bangladesh-specific CAMEO event codes and GKG themes from GDELT (Leetaru and Schrodt, 2013) between 2023 and 2025.

The resulting ontology initially comprised 16 top-level event categories and 70 sub-types. After annotation, event types with fewer than 50 instances were removed to improve evaluation reliability, yielding a final ontology of 14 main event categories and 40 sub-types (Appendix A). Definitions and representative examples for all event types and subtypes are provided in Appendix C.

3.2.1 Data Collection

News Text To construct the clean text corpus, we utilized the Bangla News Article Dataset (BNAD) (Saad et al., 2024), which contains category-wise collections of news articles sourced from Bangla Online Newspapers. We collected articles from three major Bangladeshi newspapers: Dhaka Tribune Bangla, The Daily Ittefaq, and Daily Janakantha. Articles were sampled from categories aligned with the event categories in our ontology to ensure balanced coverage across diverse domains. Coverage of topics in the dataset are described in Appendix A.

ASR Transcript To construct our real-world noisy ASR corpus, we curated publicly available news broadcasts from the official YouTube channels of three of the most popular Bangladeshi television networks: Jamuna TV, Somoy TV, and Independent Television. We extracted the automatically generated Bangla speech transcripts associated with these broadcasts. Utilizing these ASR transcripts provides a naturally occurring source of noisy Bangla text. This data inherently captures colloquial vocabulary, dialectal verb inflections, homophone substitutions, and transcription errors, accurately reflecting the morphological and lexical challenges encountered by event extraction systems in real-world spoken media.

3.3 Dataset Quality Control

Annotation Procedure We developed detailed annotation guidelines based on the proposed ontology, including event definitions, representative trigger examples, boundary annotation instructions, and common edge cases. Six annotators were selected through a screening test based on these guidelines. Annotation was conducted using the academic version of Label Studio (HumanSignal, 2020).

Inter-Annotator Agreement Each sentence was independently annotated by two annotators. Inter-annotator agreement, measured using Cohen's κ (Cohen, 1960), was 0.72.

Adjudication and Quality Control Disagreements were resolved through peer discussion followed by senior annotator adjudication. Sentences with unresolved ambiguity were excluded, resulting in the removal of approximately 18% of annotated sentences.

3.4 Dataset Statistics

In total, the finalized benchmark corpus comprises 9,979 annotated sentences, including 5,320 sentences from news articles and 4,659 sentences from ASR transcripts, containing a total of 8,041 event mentions. Dataset statistics are summarized in Table 1.

Source	# Annotated Sentences	# Event Mentions	Avg. Words /Instance
News Article	5320	4406	14.1
ASR Transcript	4659	3635	11.5
Total	**9979**	**8041**	**12.9**

Table 1: Dataset Statistics

3.5 Event Class Distribution

Figure 3 illustrates the distribution of event instances across the 40 event subtypes in the News

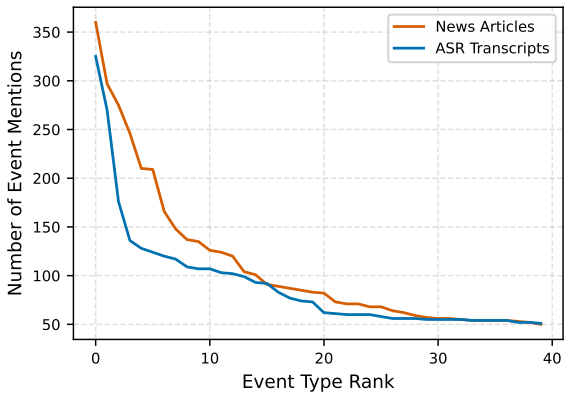


Figure 3: Frequency distribution of the 40 event subtypes in the News and ASR corpora, showing the long-tail nature of both datasets. Rankings are computed independently for each corpus.

Text and ASR Transcript corpora. Both subsets exhibit similar long-tail distributions, reflecting our topic-balancing effort. Detailed class distributions are provided in Figure 8 of Appendix A.

3.6 Orthographic Noise Test Data

Although our ASR corpus captures errors arising from acoustic ambiguity, pronunciation variation, and missing or incorrectly recognized tokens, it does not reflect the lexical and character-level spelling errors commonly found in user-generated Bangla text. These errors are primarily caused by phonetic substitutions, keystroke errors, spelling mistakes, and incorrect handling of *Juktakkhar* (consonant conjuncts). Consequently, robustness to ASR noise does not necessarily imply robustness to orthographic noise.

Rather than collecting and re-annotating a large corpus of naturally occurring spelling errors, we simulate orthographic noise using the algorithm of Sifat et al. (2020). The algorithm was developed through an analysis of real Bengali typing patterns, including phonetic substitutions, frequently mispressed key clusters, insertion errors, and *Juktakkhar*-specific rules. As a result, it produces realistic Bangla spelling errors while preserving the original event annotations.

To evaluate robustness under increasing corruption, we apply the algorithm to the Clean Test set at Word Corruption Rates (WCR), defined as $x \in \{10, 20, 30, 40\}$. At each level, approximately $x\%$ of the words are randomly selected and transformed using the error generation algorithm. Corruption is applied cumulatively, such that each higher WCR retains all previously corrupted words while introducing errors into additional randomly selected words.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Splits We evaluate models on three test sets: **Clean Test**, **ASR Test**, and **Ortho Test**, the latter generated from the Clean Test set by injecting orthographic noise at varying Word Corruption Rates (WCR). Since sentences may contain multiple event triggers following a long-tail distribution, we partition the data using iterative multi-label stratification (Sechidis et al., 2011; Szymaski and Kajdanowicz, 2017) to preserve class distributions across splits. The Clean corpus is divided into train/validation/test (70/15/15), while the ASR corpus is divided into train/validation/test (70/10/20). Table 2 summarizes the resulting dataset sizes.

Dataset	Train & Val	Test
News Text	4504	816
ASR Transcripts	3705	954
Combined (Clean + ASR)	8209	–
Ortho Test	–	816†

Table 2: Raw sentence counts across dataset splits. †Indicates the count per discrete WCR level

Models For the encoder-only baselines, we employ BanglaBERT (Base: 110M, Large: 335M) (Bhattacharjee et al., 2022) and XLM-RoBERTa (Base: 270M, Large: 550M) (Conneau et al., 2020). BanglaBERT represents the state-of-the-art monolingual pretrained model for Bangla NLP, while XLM-R provides a strong multilingual baseline.

For the generative LLM baselines, we evaluate several open-weight LLMs across different parameter scales and multilingual support. Specifically, we consider models from the Llama family, including Llama 3.1 8B IT and Llama 3.2 1B IT (Dubey et al., 2024), together with Google's Gemma 3 4B IT and Gemma 3 1B IT (Gemma Team et al., 2025).

Model Training Encoder-only models are trained for token classification, whereas decoder-only LLMs are instruction-tuned for event detection. We employ targeted negative sampling (NS), where each positive event query is

augmented with five randomly selected negative event types (1:5 ratio), providing contrastive supervision for distinguishing event types and suppressing hallucinated predictions (Srivastava et al., 2025). To evaluate the effect of annotation guidelines, we consider two prompting settings: **w/ Guide**, where event definitions are embedded as Python docstrings, and **w/o Guide**, where only the event type name is provided. Detailed prompt templates and training hyperparameters are provided in Appendix B.1 and B.2.

Evaluation Protocol Following Srivastava et al. (2025), we adopt a one-vs-all inference strategy for decoder-only LLMs, pairing each test sentence with every event type independently and aggregating predictions across all 40 event types. This formulation provides explicit supervision for absent event types, reducing hallucinated event predictions while enabling a fair comparison with encoder-only token classification models.

Evaluation Metric We report Macro-F1 as the primary metric because it equally weights all event types regardless of class frequency, making it more appropriate for our long-tailed event distribution than Micro-F1 (Al Monsur et al., 2026).

4.2 Evaluation of Language Models

LLMs exhibit greater robustness to noise than encoder-based models. To quantify robustness, we measure the performance degradation due to noise as $\Delta\text{F1} = \text{F1}_{\text{Clean}} - \text{F1}_{\text{Noisy}}$, where $\text{F1}_{\text{Noisy}} = \frac{1}{2}(\text{F1}_{\text{ASR}} + \text{F1}_{\text{Ortho-Avg}})$ averages performance on real-world ASR transcripts and simulated orthographic corruption.

Figure 4 reveals a clear distinction between encoder-based models and generative LLMs. Although encoder models achieve higher clean-text performance, they are substantially more sensitive to noise. For example, BanglaBERT Large attains the highest clean Macro F1 (68.9) but drops by 14.6 points under noisy conditions, while XLM-R Large degrades by 11.4 points. In contrast, LLMs remain considerably more robust. Llama 3.1 8B IT (w/ Guide) achieves a clean Macro F1 of 57.6 with only a 2.7 point drop, and Gemma 3 4B IT (w/ Guide) degrades by just 4.0 points.

Instruction tuning w/ Guide improves performance but yields inconsistent reductions in clean-to-noisy performance drop As shown in Figure 5 and Table 3, instruction tuning w/ Guide consistently improves Macro F1 on both the Clean and ASR test sets across all evaluated LLM families, despite training exclusively on clean news text. For example, Gemma 3 4B IT improves from 55.2 to 60.6 Macro F1 on the Clean Test set, with similar gains across other models. However, its effect on robustness is inconsistent. For Gemma 3 4B IT and Llama 3.2 1B IT, the average performance drop (ΔF1) increases from 2.9 to 4.0 and from 4.9 to 6.4 points, respectively. In contrast, Llama 3.1 8B IT benefits substantially, reducing its degradation from 6.9 to 2.7 points.

Noise combined training enhances encoder robustness Training on a mixture of clean and noisy ASR data improves robustness for both encoder-based models and generative LLMs, although the gains differ across architectures. Encoder models benefit most: BanglaBERT Large reduces its average performance drop (ΔF1) from 14.6 to 7.9 points, while XLM-R Large improves from 11.4 to 5.3 points, with little change in clean-test performance. In contrast, generative LLMs improve both clean performance and robustness. For example, Gemma 3 4B IT (w/ Guide) reduces its average performance degradation from 4.0 to 3.3 points, while Llama 3.1 8B IT (w/ Guide) reduces its degradation from 2.7 to 1.0 points, alongside an improvement in its clean-test score. Overall, combined training is particularly beneficial for encoder architectures, substantially narrowing the robustness gap with generative LLMs.

Model scaling benefits decoder-only LLMs more than encoders Parameter scaling reveals a clear architectural divergence in robustness. For decoder-only models, larger models consistently exhibit smaller noise-induced performance drops. For example, scaling from Llama 3.2 1B IT to Llama 3.1 8B IT reduces the average Macro-F1 degradation from 6.4 to 2.7 points under clean training (w/ Guide). A similar trend is observed for the Gemma family, where the 4B model consistently outperforms its 1B counterpart in robustness. In contrast, scaling encoder models from Base to Large improves clean-text performance but does not consistently improve robustness. Under clean training, BanglaBERT-Large and XLM-R-Large incur larger average performance drops than their Base counterparts (14.6 vs. 13.3 and 11.4 vs. 10.4 points, respectively).

Architecture	Model	Training Condition	Macro F1				
			Clean Test	ASR Test	Ortho Test (Avg.)	Δ_{ASR}	Δ_{Ortho}
Decoder-Only	Gemma 3 4B IT	Clean w/ Guide	60.57	59.53	53.66	1.04	6.91
		Clean w/o Guide	55.19	53.94	50.58	1.25	4.61
		Combined w/ Guide	65.60	65.43	59.14	0.17	6.46
	Gemma 3 1B IT	Clean w/ Guide	52.27	48.81	45.11	3.46	7.16
		Clean w/o Guide	48.88	45.75	44.31	3.13	4.57
		Combined w/ Guide	59.07	54.58	50.40	4.49	8.67
	Llama 3.1 8B IT	Clean w/ Guide	57.58	55.23	54.57	2.35	3.01
		Clean w/o Guide	55.27	46.92	49.86	8.35	5.41
		Combined w/ Guide	64.95	66.39	61.60	-1.44	3.35
	Llama 3.2 1B IT	Clean w/ Guide	54.64	48.07	48.38	6.57	6.26
		Clean w/o Guide	47.03	43.89	40.42	3.14	6.61
		Combined w/ Guide	56.37	59.33	51.93	-2.96	4.44
Encoder-Only	BanglaBERT Base	Clean	67.35	56.79	51.35	10.56	16.00
		Combined	68.49	68.58	52.02	-0.09	16.47
	BanglaBERT Large	Clean	68.91	58.34	50.25	10.57	18.66
		Combined	68.84	69.38	52.41	-0.54	16.43
	XLM-R Base	Clean	67.35	55.46	58.54	11.89	8.81
		Combined	68.29	66.20	58.80	2.09	9.49
	XLM-R Large	Clean	70.61	56.92	61.43	13.69	9.18
		Combined	71.16	69.34	62.33	1.82	8.83

Table 3: Macro-F1 performance of encoder and decoder architectures across clean and noisy test sets. Δ_{ASR} and Δ_{Ortho} denote the absolute performance degradation relative to the clean test set (Clean − Noisy). Smaller Δ values indicate greater robustness, while negative values indicate improved performance on the noisy test set compared with the clean baseline.

Progressive orthographic noise induces monotonic performance decline, modulated by training strategy. As Word Corruption Rate (WCR) increases from 0% to 40%, both generative LLMs and encoder-based models exhibit a monotonic decline in Macro F1 (Figure 5). However, the rate of degradation varies across architectures and training strategies. Combined training consistently improves robustness by flattening the degradation curve, with the largest gains observed for encoder models. For generative LLMs, combined training also improves absolute performance across most corruption levels. Across all evaluated model families, w/ Guide variants consistently outperform their w/o Guide counterparts on both clean and corrupted inputs.

LLMs excel under trigger corruption, while encoders remain competitive under context corruption. To complement the aggregate analysis, we perform paired McNemar's exact tests at $\text{WCR} = 40\%$, the most challenging evaluation setting. Predictions are partitioned according to whether the event trigger itself is corrupted (where surrounding context may also contain corruption) or whether only the surrounding context is corrupted while the trigger remains intact. Statistical significance is computed using strict trigger identification.

Figure 6 reveals a clear distinction between the two settings. Under trigger corruption, decoder-only LLMs, particularly larger models, significantly outperform encoder-based architectures, demonstrating greater robustness to corrupted trigger mentions. In contrast, under context corruption, the performance gap narrows substantially. Most comparisons are no longer statistically significant, and among the significant cases, XLM-R frequently outperforms the smaller LLMs. These results suggest that the primary advantage of decoder-only LLMs lies in recovering corrupted trigger mentions, whereas encoder models remain highly effective when only the surrounding context is noisy.

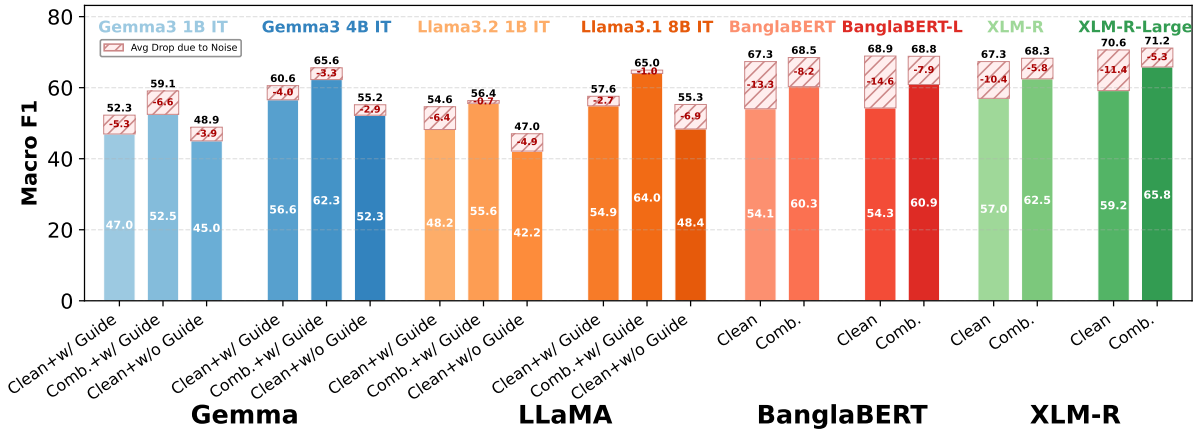


Figure 4: Macro-F1 decomposition across model architectures and training settings. Bar height denotes Clean Test Macro-F1; the solid region shows average performance retained across ASR and orthographic test sets, and the hatched region the corresponding performance drop.

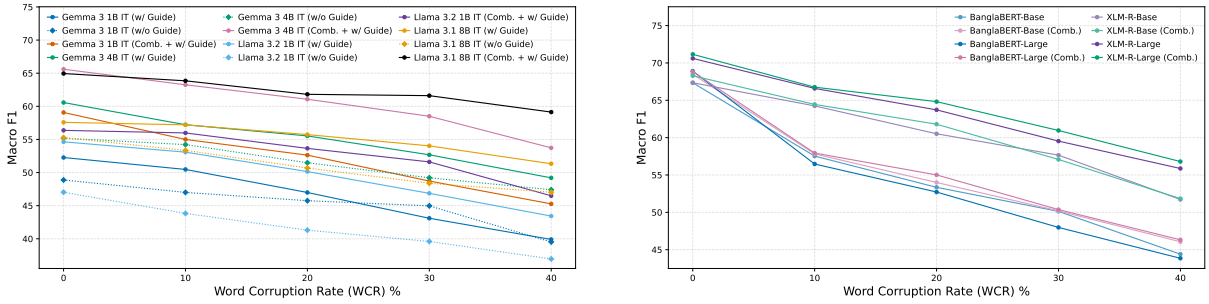


Figure 5: Comparison of performance degradation between Generative LLMs (left) and Encoder architectures (right) under varying levels of orthographic noise.

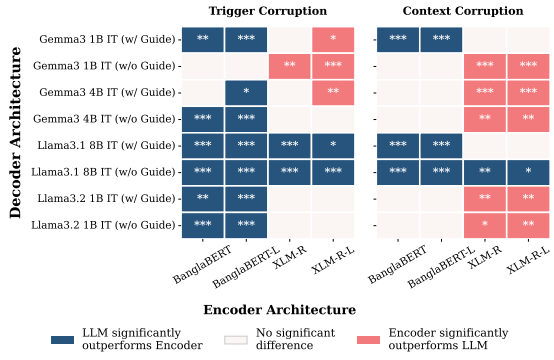


Figure 6: Pairwise McNemar's significance tests at WCR $= 40\%$. Left: Trigger Corruption (with accompanying context corruption). Right: Context-Only corruption (Trigger intact). Significance levels: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

5 Conclusion

We presented the first comprehensive study of language model robustness for Bangla event detection under real-world ASR transcription errors and simulated orthographic noise. We introduced a generalized Bangla news event ontology and released a dataset of 9,979 human-annotated sentences spanning clean news, ASR transcripts, and synthetic orthographic corruption. Our experiments reveal a clear architectural trade-off: encoder-only models achieve stronger performance on clean text but degrade substantially under ASR and orthographic noise, whereas decoder-only LLMs are considerably more robust, particularly when event triggers are corrupted. We further show that instruction tuning with annotation guidelines consistently improves overall performance but provides limited gains in robustness, while combined training on clean and noisy text substantially improves encoder robustness. These findings highlight the importance of evaluating event detection beyond clean-text benchmarks and establish strong baselines for robust event detection in low-resource settings.

Limitations

Although we carefully designed our experiments to ensure a fair comparison between encoder-based models and decoder-only LLMs, our study has several limitations.

Strict Span-Based Evaluation. Following standard event detection benchmarks, we evaluate using Strict Macro-F1, which requires an exact match between predicted and gold trigger spans and event types. This metric measures precise trigger identification, the primary objective of event detection, and enables direct comparison across encoder-only and decoder-only models. Because decoder-only LLMs generate trigger mentions autoregressively, they may occasionally produce semantically appropriate mentions with slightly different surface boundaries, which are counted as incorrect under the strict criterion. Accordingly, our results should be interpreted as reflecting exact trigger extraction rather than semantic equivalence of generated mentions.

Domain and Task Scope. This study is restricted to the news domain, using datasets constructed from formal news articles and their same-domain ASR transcripts. Consequently, the reported robustness characteristics may not generalize to general-domain event detection, which may involve other event types and exhibit substantially different linguistic properties and noise distributions. Furthermore, our investigation is limited to event detection, namely identifying event trigger mentions and classifying their event types, as annotating event arguments, participant roles, temporal expressions, and other event attributes would require substantial annotation effort and computational resources. Future research should investigate whether the robustness trends observed for event detection extend to argument extraction and complete event extraction pipelines.

Ethical Considerations

Annotators were compensated for their work at a rate consistent with local standards for similar annotation tasks. Prior to participation, annotators were informed about the purpose of the study and how their contributions would be used in the dataset. All news articles and Automatic Speech Recognition (ASR) transcripts were collected from publicly available channels. The dataset was compiled strictly for non-commercial, academic research purposes under standard fair use principles. Any named entities or individuals present in the text are public figures or subjects of public record, and no sensitive Personally Identifiable Information (PII) of private citizens was targeted or exposed.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Abdullah Al Monsur, Nitesh Vamshi Bommisetty, and Gene Louis Kim. 2026. Event Detection with a Context-Aware Encoder and LoRA for Improved Performance on Long-Tailed Classes. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 5985–6003, Rabat, Morocco. Association for Computational Linguistics.

Iftakhar Ali Khandokar, Abdullah All Tanvir, Md. Saddam Hossain Mukta, and Swakkhar Shatabda. 2025. Temporal, Demographic, and Geographical Analysis of Violent Events in Bangla News Media Using NLP Techniques. *Human-Centric Intelligent Systems*, 5(1):90–102.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. BanglaBERT: Language Model Pretraining and Benchmarks for Low-Resource Language Understanding Evaluation in Bangla. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Jacob Cohen. 1960. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Bhargav Dave, Surupendu Gangopadhyay, Prasenjit Majumder, Pushpak Bhattacharya, Sudeshna Sarkar, and Sobha Lalitha Devi. 2021. FIRE 2020 EDNIL Track: Event Detection from News in Indian Languages. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, pages 25–28, New York, NY, USA. Association for Computing Machinery.

Noyon Dey, Md. Sazzadur Rahman, Motahara Sabah Mredula, A. S. M. Sanwar Hosen, and In-Ho Ra. 2021. Using Machine Learning to Detect Events on the Basis of Bengali and Banglish Facebook Posts. *Electronics*, 10(19):2367.

Abhimanyu Dubey, Abhinav Jauhri, and et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-Sentence Argument Linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Kazi Toufique Elahi, Tasnuva Binte Rahman, Shakil Shahriar, Samir Sarker, Md Tanvir Rouf Shawon, and G. M. Shahariar. 2024. A Comparative Analysis of Noise Reduction Methods in Sentiment Analysis on Noisy Bangla Texts. *Preprint*, arXiv:2401.14360.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 193 others. 2025. Gemma 3 Technical Report. *arXiv preprint*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using Cross-Entity Inference to Improve Event Extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA. Association for Computational Linguistics.

Md. Mithun Hossain, Sanjara, Md. Shakil Hossain, Sudipto Chaki, Md. Saifur Rahman, and A B M Shawkat Ali. 2025. MaskNet: Enhancing Crime Event Detection with Feature Masking and Dynamic Attention. In *2025 2nd International Conference on Next-Generation Computing, IoT and Machine Learning (NCIM)*, pages 1–6.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. *Preprint*, arXiv:2106.09685.

Kuan-Hao Huang, I-Hung Hsu, Tanmay Parekh, Zhiyu Xie, Zixuan Zhang, Prem Natarajan, Kai-Wei Chang, Nanyun Peng, and Heng Ji. 2024. TextEE: Benchmark, Reevaluation, Reflections, and Future Challenges in Event Extraction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12804–12825, Bangkok, Thailand. Association for Computational Linguistics.

HumanSignal. 2020. Label Studio: Data labeling software. Available at `https://labelstud.io`.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A Dataset for Analysing Sentiment on Noisy Bangla Texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Iftakhar Ali Khandokar, Imtiaz Mamun, Tasmia Ishrat Alam Chadni, Zubair Ahmed Anas, and Swakkhar Shatabda. 2020. Event Detection and Knowledge Mining from Unlabelled Bengali News Articles. In *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, pages 1–6, Bangladesh. IEEE.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2009. Overview of BioNLP'09 Shared Task on Event Extraction. In *Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task*, pages 1–9, Boulder, Colorado. Association for Computational Linguistics.

Duong Le and Thien Huu Nguyen. 2021. Fine-Grained Event Trigger Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2745–2752, Online. Association for Computational Linguistics.

Kalev Leetaru and Philip Schrodt. 2013. Gdelt: Global data on events, language, and tone, 1979-2012. In *International Studies Association Annual Conference*, San Francisco, CA.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint Event Extraction via Structured Prediction with Global Features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. *Preprint*, arXiv:1711.05101.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A Light-Weight Transformer-based Toolkit for Multilingual Natural Language Processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 80–90, Online. Association for Computational Linguistics.

Chaoxu Pang, Yixuan Cao, Qiang Ding, and Ping Luo. 2023. Guideline Learning for In-Context Information Extraction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15372–15389, Singapore. Association for Computational Linguistics.

Siddharth Patwardhan and Ellen Riloff. 2009. A Unified Model of Phrasal and Sentential Evidence for Information Extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 151–160, Singapore. Association for Computational Linguistics.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022. MINION: A Large-Scale and Diverse Dataset for Multilingual Event Detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2286–2299, Seattle, United States. Association for Computational Linguistics.

Asif Mohammed Saad, Umme Niraj Mahi, Md. Shahidul Salim, and Sk Imran Hossain. 2024. Bangla news article dataset. *Data in Brief*, 57:110874.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2024. GoLLIE: Annotation Guidelines improve Zero-Shot Information-Extraction. *Preprint*, arXiv:2310.03668.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. *Machine Learning and Knowledge Discovery in Databases*, pages 145–158.

Omar Sharif, Joseph Gatto, Madhusudan Basak, and Sarah Masud Preum. 2024. Explicit, Implicit, and Scattered: Revisiting Event Extraction to Capture Complex Arguments. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12061–12081, Miami, Florida, USA. Association for Computational Linguistics.

Md Habibur Rahman Sifat, Chowdhury Rafeed Rahman, Mohammad Rafsan, and Md Hasibur Rahman. 2020. Synthetic Error Dataset Generation Mimicking Bengali Writing Pattern. *Preprint*, arXiv:2003.03484.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary Event Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.

Saurabh Srivastava, Sweta Pati, and Ziyu Yao. 2025. Instruction-Tuning LLMs for Event Extraction with Annotation Guidelines. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13055–13071, Vienna, Austria. Association for Computational Linguistics.

Piotr Szymaski and Tomasz Kajdanowicz. 2017. A network perspective on stratification of multi-label data. In *Proceedings of the First International Workshop on Learning with Imbalanced Domains: Theory and Applications*, volume 74 of *Proceedings of Machine Learning Research*, pages 22–35, ECML-PKDD, Skopje, Macedonia. PMLR.

Samia Touileb, Jeanett Murstad, Petter Mæhlum, Lubos Steskal, Lilja Charlotte Storset, Huiling You, and Lilja Øvrelid. 2024. EDEN: A Dataset for Event Detection in Norwegian News. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5495–5506, Torino, Italia. ELRA and ICCL.

Walker, Christopher, Strassel, Stephanie, Medero, Julie, and Maeda, Kazuaki. 2006. ACE 2005 Multilingual Training Corpus.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

Xingyao Wang, Sha Li, and Heng Ji. 2023. Code4Struct: Code Generation for Few-Shot Event Structure Prediction. *Preprint*, arXiv:2210.12810.

Feng Yao, Chaojun Xiao, Xiaozhi Wang, Zhiyuan Liu, Lei Hou, Cunchao Tu, Juanzi Li, Yun Liu, Weixing Shen, and Maosong Sun. 2022. LEVEN: A Large-Scale Chinese Legal Event Detection Dataset. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 183–201, Dublin, Ireland. Association for Computational Linguistics.

A Data Collection and Ontology Development

A.1 Topic Coverage

To ensure comparable topic-coverage across clean text and ASR transcripts, we curated both corpora from the same set of news domains. For the clean text corpus, we selected articles from the BNAD (Saad et al., 2024) dataset, specifically using content from three major Bangladeshi newspapers: *Dhaka Tribune Bangla*, *The Daily Ittefaq*, and *Daily Janakantha*. We focused on seven major news domains:

- রাজনীতি (*Politics*)
- অর্থনীতি (*Economy*)
- স্বাস্থ্য (*Health*)
- শিক্ষা (*Education*)
- প্রযুক্তি/টেক/বিজ্ঞান ও প্রযুক্তি (*Technology/Science*)
- বাণিজ্য (*Commerce*)
- চাকরি (*Jobs*)

For the ASR corpus, we curated publicly available Bangla news broadcasts from the official YouTube channels of *Jamuna TV*, *Somoy TV*, and *Independent Television*, and extracted their automatically generated Bangla speech transcripts. To mirror the clean text corpus, we collected broadcasts from category-specific playlists corresponding to the same seven news domains whenever available.

A.2 Ontology Development

To construct a generalized Bangla news event ontology, we analyzed a large-scale corpus of Bangladeshi news using the Global Database of Events, Language, and Tone (GDELT) through Google BigQuery and queried the Global Knowledge Graph (GKG) to extract historical event records related to Bangladesh from 2023 to 2025 (Figure 7). Based on this corpus analysis, we identified event phenomena that are prevalent in Bangladeshi news and introduced locally relevant event types and subtypes to complement the widely adopted ACE 2005 ontology. This process ensured compatibility with established event types while capturing events that are underrepresented or absent in ACE 2005 but frequently occur in Bangladeshi news. The resulting ontology comprises 40 event subtypes organized into generalized event categories. The complete list of event types and subtypes, together with their sources, is presented in Table 4. The distribution of annotated event instances across the News and ASR corpora is shown in Figure 8.

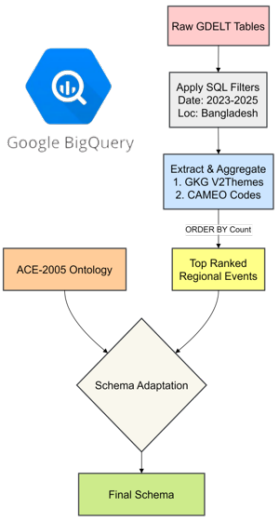


Figure 7: Overview of the ontology construction process. We analyze historical Bangladeshi news events from GDELT using Google BigQuery and the Global Knowledge Graph (GKG), adapt relevant event types, and integrate them with ACE 2005 to construct a generalized Bangla news event ontology.

Table 4: Event types and sub types in the dataset

Event Type	**Sub Type**	**Source**
Contact	Meet	ACE 2005
	Phone-Write	ACE 2005
Conflict	Attack	ACE 2005
	Demonstrate	ACE 2005
Crime	Commit-Blue-Collar-Crime	GDELT
	Commit-White-Collar-Crime	GDELT
Disaster	Occur-Man-Made-Disaster	GDELT
	Occur-Natural-Disaster	GDELT
Festival	Celebrate-Cultural-Festival	GDELT
	Observe-Religious-Festival	GDELT
Governance	Approve	GDELT
	Ban	GDELT
	Decide	GDELT
	Gazette	GDELT
	Investigate	GDELT
	Reform	GDELT
	Support	GDELT
Health	Outbreak	GDELT
	Serve-Patients	GDELT
Justice	Arrest-Jail	ACE 2005
	Charge-Indict	ACE 2005
	Deliver-Verdict	ACE 2005
	Sue	ACE 2005
	Trial-Hearing	ACE 2005
Life	Die	ACE 2005
	Injure	ACE 2005
Movement	Transport	ACE 2005
Personnel	Elect	ACE 2005
	End-Position	ACE 2005
	Start-Position	ACE 2005
Socio-Economy	Disrupt	GDELT
	Graduate	GDELT
	Grow	GDELT
	Import-Export	GDELT
	Invest	GDELT
	Recruit	GDELT
	Trade	GDELT
Technology	Launch-Service	GDELT
Transaction	Transfer-Money	ACE 2005
	Transfer-Ownership	ACE 2005

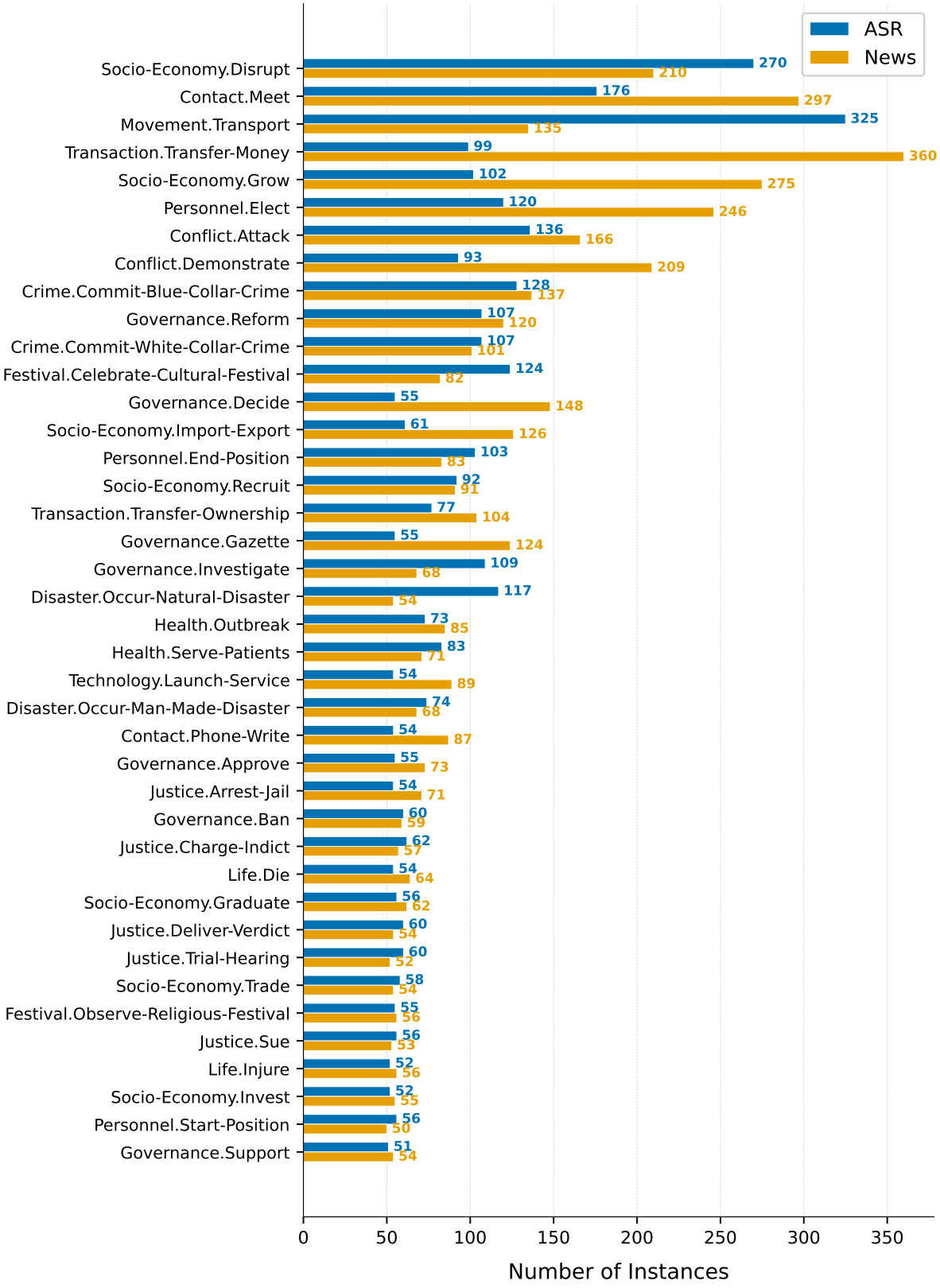


Figure 8: Distribution of event instances across the 40 event subtypes in the News and ASR corpora. Both corpora exhibit similar long-tail distributions, reflecting our topic-balancing effort despite the ASR corpus containing naturally occurring transcription artifacts and out-of-distribution vocabulary.

B Prompt Design and Model Training

B.1 Instruction Tuning Prompt Formats

The following example illustrates the structured code generation prompt under the **w/ Guide** setting, where fine-grained annotation guidelines (event definitions, trigger examples, and boundary constraints) are embedded directly within the schema representation as Python docstrings. English translations are provided in brackets for reference.

Instruction Tuning Prompt (w/ Guide)

```
# Instruction
Extract events according to the
    schema.

# Schema
@dataclass
class Injure(Life):
    """
    এই ইভেন্টটি তখন ঘটে যখন
    অনিচ্ছাকৃত/স্বেচ্ছায়/দুর্ঘটনাবশত কোন
    ব্যক্তি শারীরিকভাবে আঘাত পান।
    (This event occurs when a person
    is physically injured unintention-
    ally/voluntarily/accidentally.)

    উদাহরণ ট্রিগার: আঘাত, ক্ষতি, আহত,
    অঙ্গহানি ইত্যাদি।
    (Example triggers: injure, harm,
    injured, dismemberment, etc.)

    সতর্কতা: (Note:)
    - আঘাতপ্রাপ্ত ব্যক্তি মারা গেলে তা এই
    ইভেন্টের অন্তর্ভুক্ত হবে না।
    (If the injured person dies, it will
    not be included in this event.)
    """
    mention: str

# Sentence
text = "আহত শিক্ষার্থীকে উদ্ধার করা সম্ভব
    হয়েছে।"
        (The injured student was
            successfully rescued.)

result =

# Output
[
    Injure(mention="আহত")
        ("injured")
]
```

Instruction Tuning Prompt (w/o Guide)

```
# Instruction
Extract events according to the
    schema.

# Schema
@dataclass
class Attack(Conflict):
    mention: str

# Sentence
text = "খুলনা বিশ্ববিদ্যালয়ে শিক্ষার্থী
    নির্যাতনের অভিযোগে শিক্ষার্থীর সীট বাতিল
    করেছে হল কর্তৃপক্ষ"
        (The hall authority has cancelled
            the seat of a student on
            allegations of student
            torture at Khulna
            University.")

result =

# Output
[
    Attack(mention="নির্যাতনের")
        ("torture")

]
```

B.2 Training Details

Encoders Encoder-only models are fine-tuned using full-parameter supervised fine-tuning with the AdamW optimizer (Loshchilov and Hutter, 2019), a learning rate of 2×10^{-5}, weight decay of 0.01, warmup over 10% of training steps, and a maximum sequence length of 128 tokens. Training is conducted for up to 15 epochs with early stopping based on validation Macro-F1 with patience 3.

Decoders Decoder-only LLMs are adapted via Low-Rank Adaptation (Hu et al., 2021) with rank $r = 16$, scaling factor $\alpha = 32$, and dropout $p = 0$, applied to all linear projection layers. We use a uniform rank across all model sizes following Al Monsur et al. (2026), who demonstrate that ranks above 8 provide no additional benefit for event detection tasks. Models are trained for up to 5 epochs with a learning rate of 1×10^{-4}, cosine learning rate schedule, warmup ratio of 0.05, effective batch size of 64 via gradient accumulation, and early stopping on validation loss with patience 3. All experiments are conducted on a single NVIDIA A100 GPU.

C Comprehensive Event Ontology Guidelines

Table 5: Detailed definitions and examples for all 40 event subtypes.

Event Type	Sub Type	Definition	Example
Contact	Meet	যখন দুই বা ততোধিক ব্যক্তি, দল বা প্রতিষ্ঠানের প্রতিনিধিরা সামনাসামনি সাক্ষাৎ, বৈঠক, সভা বা সম্মেলনে অংশগ্রহণ করেন। (When two or more individuals, groups, or organizational representatives participate in a face-to-face meeting, conference, gathering, or discussion.)	সমসাময়িক বিষয় নিয়ে জরুরি এ সংবাদ **সম্মেলন** ডাকা হয়। (An emergency press **conference** was called regarding the current situation.)
	Phone-Write	যখন দুই বা ততোধিক ব্যক্তি, দল বা প্রতিষ্ঠানের প্রতিনিধিরা সামনাসামনি সাক্ষাতের পরিবর্তে ফোন, ইমেইল, চিঠি বা অন্যান্য লিখিত বার্তার মাধ্যমে যোগাযোগ বা আলোচনা করেন। (When two or more individuals, groups, or organizational representatives communicate or discuss through phone calls, emails, letters, or other written messages instead of meeting face-to-face.)	বিএনপির ভারপ্রাপ্ত চেয়ারম্যান চার নেতাকে **ফোন** করে নির্বাচনী প্রস্তুতি সম্পর্কে জানতে চান। (The Acting Chairman of the BNP **called** four party leaders to inquire about the election preparations)
Conflict	Attack	যখন কোনো ব্যক্তি বা গোষ্ঠীর বিরুদ্ধে শারীরিক আক্রমণ, হামলা, সংঘর্ষ, যুদ্ধ, অভ্যুত্থান বা গুলিবর্ষণের মতো সহিংস ঘটনা ঘটে। আঘাতের নির্দিষ্ট মাধ্যম উল্লেখ না থাকলেও, যদি ঘটনাটি শারীরিক সহিংসতা বা সন্ত্রাসী কর্মকাণ্ড নির্দেশ করে, তবে সেটি *Attack* হিসেবে বিবেচিত হবে। (When an individual or group is involved in a physically violent incident such as an attack, assault, clash, armed conflict, coup, or shooting. The specific means of injury need not be explicitly stated, provided the event describes physical violence or terrorist-like activity.)	ব্রাহ্মণবাড়িয়ায় আধিপত্য দ্বন্দ্বে দুই পক্ষের রক্তক্ষয়ী **সংঘর্ষ** হয়। (A deadly **clash** occurred between two rival groups over dominance in Brahmanbaria.)
	Demonstrate	যখন কোনো ব্যক্তি বা জনগোষ্ঠী কোনো দাবি, অধিকার বা নির্দিষ্ট উদ্দেশ্য বাস্তবায়নের লক্ষ্যে সংগঠিতভাবে বিক্ষোভ, আন্দোলন, মানববন্ধন, ধর্মঘট, হরতাল বা অবরোধে অংশগ্রহণ করে। (When an individual or group collectively participates in a protest, demonstration, human chain, strike, hartal, or blockade to express demands, grievances, or support for a particular cause.)	জনগণ ২৯ তারিখের **হরতাল** স্বপ্রণোদিত হয়ে পালন করেছে। (The public voluntarily observed the **hartal** on the 29th.)

Event Type	Sub Type	Definition	Example
Crime	Commit-Blue-Collar-Crime	যখন নিম্ন আয়ের ব্যক্তি বা গোষ্ঠী কর্তৃক সংঘটিত প্রচলিত অপরাধ, যেমন হত্যা, ডাকাতি, ছিনতাই, অপহরণ, ধর্ষণ, চোরাচালান, মানবপাচার বা অগ্নিসংযোগের ঘটনা স্পষ্টভাবে উল্লেখ থাকে। অপরাধের ধরন অবশ্যই সুস্পষ্টভাবে উল্লেখ থাকতে হবে; শুধুমাত্র সংঘর্ষ, দ্বন্দ্ব, মারামারি বা গুলিবর্ষণের কথা বলা থাকলে সেটি *Conflict.Attack* হবে। (When a conventional crime committed by an individual or group of lower socioeconomic background, such as murder, robbery, mugging, kidnapping, rape, smuggling, human trafficking, or arson, is explicitly mentioned. The type of crime must be clearly specified; if the text only describes a clash, conflict, fight, or shooting without explicitly identifying a crime, it should be Conflict.Attack.)	রাজধানীতে **ছিনতাই** রোধে শেষ রাতে টহল বাড়ানোর নির্দেশনা দেওয়া হয়েছে। (Authorities have instructed increased late-night patrols in the capital to prevent **mugging**.)
	Commit-White-Collar-Crime	যখন কোনো ব্যক্তি বা গোষ্ঠী দুর্নীতি, ঘুষ, প্রতারণা, জালিয়াতি, আত্মসাৎ, সাইবার অপরাধ, মানি লন্ডারিং বা অর্থ পাচারের মতো আর্থিক বা পেশাগত অপরাধে জড়িত থাকে। তবে প্রতারণা বা জালিয়াতির মতো অপরাধ সমাজের নিম্ন আয়ের ব্যক্তি কর্তৃক সংঘটিত হলেও *Commit-White-Collar-Crime* হিসেবে বিবেচিত হবে। (When an individual or group commits a financial or occupational crime such as corruption, bribery, fraud, forgery, embezzlement, cybercrime, money laundering, or illicit financial transfer. Fraud and forgery committed by individuals from any socioeconomic background should also be annotated as Commit-White-Collar-Crime.)	তাঁর বিরুদ্ধে চেক **জালিয়াতির** অভিযোগে একাধিক মামলা হয়েছে। (Multiple cases have been filed against him on charges of check **forgery**.)
Disaster	Occur-Man-Made-Disaster	যখন মানবসৃষ্ট বা অনিচ্ছাকৃত দুর্যোগ, যেমন অগ্নিকাণ্ড, বিস্ফোরণ, সড়ক দুর্ঘটনা, ভবন ধস বা অনুরূপ দুর্ঘটনা সংঘটিত হয়। (When a human-induced or accidental disaster, such as a fire, explosion, road accident, building collapse, or similar incident, occurs.)	শুক্রবার দুপুর সোয়া দুইটায় ঢাকার হাজারীবাগে চামড়ার গুদামে **আগুন লাগে**। **(A fire broke out at a leather warehouse in Hazaribagh, Dhaka, at around 2:15 p.m. on Friday.)**
	Occur-Natural-Disaster	যখন বন্যা, ঘূর্ণিঝড়, ভূমিকম্প, ভূমিধস, জলোচ্ছ্বাস, নদীভাঙন বা অন্যান্য প্রাকৃতিক দুর্যোগ সংঘটিত হয় বা তার প্রভাবের কথা উল্লেখ থাকে। (When a natural disaster, such as a flood, cyclone, earthquake, landslide, storm surge, river erosion, or similar natural hazard, occurs or its impact is described.)	সুরমা নদীর পানিতে **প্লাবিত** হয়েছে নদীতীরবর্তী গ্রামগুলো। (The riverside villages have been **flooded** by the water of the Surma River.)

Event Type	Sub Type	Definition	Example
Festival	Celebrate-Cultural-Festival	যখন কোনো সাংস্কৃতিক উৎসব, জাতীয় দিবস, মেলা বা ঐতিহ্যবাহী অনুষ্ঠান উদযাপন, আয়োজন বা পালন করা হয়। (When a cultural festival, national day, fair, or traditional celebration is organized, observed, or celebrated.)	দুই দেশই কূটনৈতিক সম্পর্কের সুবর্ণজয়ন্তী **উদযাপন** করেছে। (Both countries **celebrated** the golden jubilee of their diplomatic relations.)
	Observe-Religious-Festival	যখন কোনো ধর্মীয় উৎসব, ধর্মীয় অনুষ্ঠান বা ধর্মীয় আচার-অনুষ্ঠান, যেমন ঈদ, দুর্গাপূজা, বুদ্ধ পূর্ণিমা, বড়দিন বা অনুরূপ ধর্মীয় দিবস পালন বা উদযাপন করা হয়। (When a religious festival, ceremony, or observance, such as Eid, Durga Puja, Buddha Purnima, Christmas, or similar religious occasions, is celebrated or observed.)	**বৌদ্ধ পূর্ণিমা** উপলক্ষে আজ শান্তি শোভাযাত্রা অনুষ্ঠিত হয়েছে। (A peace procession was held today to mark **Buddha Purnima**.)
Governance	Approve	যখন কোনো সরকার, সরকারি সংস্থা বা প্রতিষ্ঠান কোনো আইন, প্রকল্প, ঋণ, বাজেট, বরাদ্দ বা নীতিগত সিদ্ধান্ত আনুষ্ঠানিকভাবে অনুমোদন করে। (When a government, public agency, or organization formally approves a law, project, loan, budget, allocation, or policy.)	মাধ্যমিক শিক্ষায় ৩০ কোটি ডলার ঋণ **অনুমোদন** করেছে বিশ্বব্যাংক। (The World Bank **approved** a $300 million loan for secondary education.)
	Ban	যখন কোনো সরকার, সরকারি সংস্থা বা প্রতিষ্ঠান আদালতের আদেশ ব্যতীত কোনো ব্যক্তি, প্রতিষ্ঠান, কার্যক্রম, পণ্য বা সেবার ওপর নিষেধাজ্ঞা আরোপ বা কালোতালিকাভুক্ত করে। (When a government, public agency, or organization imposes a ban or blacklists a person, organization, activity, product, or service, excluding bans issued through a court order.)	বাংলাদেশের ব্যাংকগুলোর সব ধরনের যানবাহন কেনায় **নিষেধাজ্ঞা** দিয়েছে কেন্দ্রীয় ব্যাংক। (The central bank has imposed a **ban** on all vehicle purchases by banks in Bangladesh.)
	Decide	যখন কোনো সরকার, সরকারি সংস্থা, প্রতিষ্ঠান, পরিচালনা পর্ষদ বা অনুমোদিত কর্তৃপক্ষ আনুষ্ঠানিকভাবে কোনো সিদ্ধান্ত গ্রহণ করে। (When a government, public agency, organization, board of directors, or other authorized authority formally makes a decision.)	বাজুসের কার্যনির্বাহী কমিটির সদস্যদের সর্বসম্মত মতামতের ভিত্তিতে দাম বাড়ানোর **সিদ্ধান্ত** হয়েছে। (Based on the unanimous opinion of the BAJUS executive committee members, a **decision** was made to increase the price.)
	Gazette	যখন কোনো সরকার, সরকারি সংস্থা বা প্রতিষ্ঠান আনুষ্ঠানিকভাবে প্রজ্ঞাপন, গেজেট বা নোটিশ জারি করে। (When a government, public agency, or organization officially issues a gazette notification, proclamation, or formal notice.)	সোমবার এ-সংক্রান্ত **প্রজ্ঞাপন** জারি করেছে শিক্ষা মন্ত্রণালয়। (The Ministry of Education issued a **gazette notification** regarding the matter on Monday.)

Event Type	Sub Type	Definition	Example
	Investigate	যখন কোনো অভিযোগ, ঘটনা বা অনিয়মের সত্যতা যাচাইয়ের উদ্দেশ্যে সরকার, তদন্তকারী সংস্থা বা কোনো প্রতিষ্ঠান আনুষ্ঠানিকভাবে তদন্ত বা অনুসন্ধান পরিচালনা করে। (When a government, investigative authority, or organization formally conducts an investigation or inquiry to verify the facts surrounding an allegation, incident, or irregularity.)	দুর্নীতির অভিযোগে দুদক ঘটনাটি **তদন্ত** করছে। (The Anti-Corruption Commission is **investigating** the incident following allegations of corruption.)
	Reform	যখন কোনো সরকার, সরকারি সংস্থা বা প্রতিষ্ঠান জনকল্যাণের উদ্দেশ্যে কোনো নীতি, আইন, বিধিমালা বা প্রশাসনিক ব্যবস্থার প্রণয়ন, পরিবর্তন বা সংস্কার করে। (When a government, public agency, or organization formulates, revises, or reforms a policy, law, regulation, or administrative system intended to improve governance or public welfare.)	নিরাপদ খাদ্য কর্তৃপক্ষ ট্রান্সফ্যাট নিয়ন্ত্রণ নীতি **প্রণয়নের** উদ্যোগ নিয়েছে। (The Bangladesh Food Safety Authority has initiated the **formulation** of a policy to regulate trans fats.)
	Support	যখন কোনো সরকার, সরকারি সংস্থা বা প্রতিষ্ঠান কোনো ব্যক্তি, গোষ্ঠী বা প্রতিষ্ঠানকে নগদ অর্থ, ভর্তুকি, ঋণ, ত্রাণ বা অর্থমূল্যসম্পন্ন অন্যান্য সহায়তা প্রদান করে। (When a government, public agency, or organization provides financial or material assistance, such as subsidies, loans, cash, relief, or other support with monetary value, to an individual, group, or organization.)	১০ টাকা কেজি চাল কর্মসূচিতে সরকার ২ হাজার ২০০ কোটি টাকা **ভর্তুকি** দিচ্ছে। (The government is providing **subsidies** worth BDT 22 billion for the rice program that sells rice at BDT 10 per kilogram.)
Health	Outbreak	যখন কোনো সংক্রামক বা অসংক্রামক রোগের প্রাদুর্ভাব ঘটে, কোনো ব্যক্তি বা জনগোষ্ঠী আক্রান্ত হয়, রোগ শনাক্ত হয়, অথবা রোগের সংক্রমণের ঘটনা উল্লেখ থাকে। (When a disease outbreak occurs, an individual or population is infected, a disease is diagnosed, or the transmission of a disease is reported.)	সম্প্রতি দেশে ডেঙ্গু রোগের **প্রাদুর্ভাব** বৃদ্ধি পেয়েছে। (Recently, the **outbreak** of dengue has increased across the country.)
	Serve-Patients	যখন কোনো চিকিৎসক, হাসপাতাল বা স্বাস্থ্যসেবা প্রতিষ্ঠান রোগীকে চিকিৎসা, ভর্তি, পরামর্শ বা অন্যান্য স্বাস্থ্যসেবা প্রদান করে। (When a doctor, hospital, or healthcare institution provides medical treatment, hospital admission, consultation, or other healthcare services to a patient.)	বর্তমানে তিনি আইসিইউতে **চিকিৎসাধীন** রয়েছেন। (He is currently **undergoing treatment** in the intensive care unit (ICU)).

Event Type	Sub Type	Definition	Example
Justice	Arrest-Jail	যখন পুলিশ, আইনশৃঙ্খলা রক্ষাকারী বাহিনী বা অন্য কোনো রাষ্ট্রীয় কর্তৃপক্ষ আইনগতভাবে কোনো অভিযুক্ত ব্যক্তিকে গ্রেফতার বা আটক করে, অথবা আদালতের আদেশে কারাগারে প্রেরণ বা কারাদণ্ড কার্যকর করা হয়। (When a police force, law enforcement agency, or other government authority legally arrests or detains an accused person, or when an individual is imprisoned or sent to jail by court order.)	এ ঘটনায় বিএনপির আট জনকে **গ্রেফতার** করা হয়। (Eight BNP members were **arrested** in connection with the incident.)
	Charge-Indict	যখন কোনো ব্যক্তি বা প্রতিষ্ঠানের বিরুদ্ধে আনুষ্ঠানিকভাবে কোনো অপরাধ বা অনিয়মের অভিযোগ আনা হয়, অথবা তাকে অভিযুক্ত করা হয়। (When an individual or organization is formally charged, accused, or indicted for a crime or wrongdoing. The allegation does not need to be proven.)	প্রাথমিকভাবে **অভিযুক্ত** পাঁচজনকে সাময়িক বহিষ্কার করা হয়েছে। (The five individuals initially **accused** were temporarily suspended.)
	Deliver-Verdict	যখন কোনো আদালত কোনো মামলার রায়, আদেশ বা নির্দেশ প্রদান করে। (When a court delivers a verdict, judgment, or judicial order in a legal case.)	মনিপুর স্কুলের বিষয়ে আদালতের **রায়** মেনে নেওয়ার আশ্বাস দিয়েছেন প্রতিমন্ত্রী। (The State Minister assured that the court's **verdict** regarding Monipur School would be accepted.)
	Sue	যখন কোনো ব্যক্তি, প্রতিষ্ঠান বা সংস্থা আদালতে অন্য ব্যক্তি বা প্রতিষ্ঠানের বিরুদ্ধে মামলা দায়ের করে। (When an individual, organization, or institution files a lawsuit or legal case against another individual or organization.)	ভুক্তভোগী অভিযুক্তদের বিরুদ্ধে **মামলা** করেছেন। (The victim has filed a **lawsuit** against the accused.)
	Trial-Hearing	যখন কোনো আদালত কোনো মামলার আনুষ্ঠানিক শুনানি, জামিন শুনানি বা বিচারিক কার্যক্রম পরিচালনা করে। (When a court conducts a formal trial, hearing, bail hearing, or other judicial proceeding related to a legal case.)	বিশ্ববিদ্যালয়ের ক্যাম্পাসের জমি কেনা-সংক্রান্ত অর্থ আত্মসাতের মামলায় ট্রাস্টি বোর্ডের চার সদস্যের জামিন **শুনানি** রোববার অনুষ্ঠিত হবে। (The bail **hearing** for four members of the Board of Trustees in the embezzlement case related to the purchase of land for the University will be held on Sunday.)

Event Type	Sub Type	Definition	Example
Life	Die	যখন কোনো ব্যক্তি স্বাভাবিক কারণে, দুর্ঘটনায়, সহিংস ঘটনায় বা আত্মত্যাগের ফলে মৃত্যুবরণ করেন। মৃত্যুর ফলাফল নির্দেশকারী শব্দ, যেমন নিহত, মৃত, মারা যান বা প্রাণ হারান উল্লেখ থাকলেও এটি *Die* ইভেন্ট হিসেবে বিবেচিত হবে। (When a person dies due to natural causes, an accident, violence, or self-sacrifice. Resultative expressions indicating death, such as killed, dead, died, or lost one's life, should also be annotated as Die events.)	**নিহত** শাহীনের পরিবারের পাশে দাঁড়াতে নেতাকর্মীদের প্রতি আহ্বান জানানো হয়। (Party leaders and activists were urged to stand beside the family of the **deceased** Shahin.)
	Injure	যখন কোনো ব্যক্তি দুর্ঘটনা, সহিংসতা বা অন্য কোনো কারণে শারীরিকভাবে আহত বা আঘাতপ্রাপ্ত হন। (When a person sustains a physical injury due to an accident, violence, or any other cause. If the person dies as a result of the incident, the event should be annotated as Die, not Injure.)	গাছের সঙ্গে ধাক্কা লেগে বিশ্ববিদ্যালয়ের অন্তত ১৬ শিক্ষার্থী **আহত** হন। (At least 16 university students were **injured** after their vehicle collided with a tree.)
Movement	Transport	যখন কোনো ব্যক্তি, যানবাহন বা পণ্য এক স্থান থেকে অন্য স্থানে পরিবহন, যাত্রা, স্থানান্তর বা রওনা হয়। (When a person, vehicle, or goods are transported, travel, depart, or move from one location to another.)	তিনি বিমান বাংলাদেশের একটি ফ্লাইটে চিকিৎসার জন্য ভারতের চেন্নাইয়ের উদ্দেশ্যে **রওনা** দিয়েছেন। (He **departed** for Chennai, India, on a Biman Bangladesh Airlines flight for medical treatment.)
Personnel	Elect	যখন কোনো ব্যক্তি নির্বাচনের মাধ্যমে কোনো পদে নির্বাচিত হন, অথবা কোনো পদে প্রতিনিধি নির্বাচনের উদ্দেশ্যে নির্বাচন অনুষ্ঠিত বা আয়োজন করা হয়। (When a person is elected to a position through an election, or when an election is held or organized to select representatives for a position.)	ঢাকা চেম্বার অব কমার্স অ্যান্ড ইন্ডাস্ট্রির (ডিসিসিআই) সভাপতি **নির্বাচিত** হয়েছেন তিনি। (He was **elected** President of the Dhaka Chamber of Commerce and Industry.)
	End-Position	যখন কোনো ব্যক্তি চাকরি, পদ বা দায়িত্ব থেকে বরখাস্ত, পদচ্যুত, ছাঁটাই, অব্যাহতি, পদত্যাগ বা অবসরের মাধ্যমে দায়িত্ব থেকে সরে যান। (When a person leaves a job or position through dismissal, termination, removal, resignation, retirement, or the end of their tenure.)	এক নারী কর্মীকে চাকরি থেকে **বরখাস্ত** করেছে চীনা ই-কমার্স প্রতিষ্ঠান আলিবাবা। (The Chinese e-commerce company Alibaba **dismissed** a female employee from her job.)
	Start-Position	যখন কোনো ব্যক্তি কোনো পদে নিয়োগপ্রাপ্ত হন, নতুন চাকরিতে যোগদান করেন বা আনুষ্ঠানিকভাবে দায়িত্ব গ্রহণ করেন। (When a person is appointed to a position, joins a new job, or officially assumes a role or responsibility.)	একদিনের চাকরি মেলায় তিনজন প্রার্থী **নিয়োগ** পেয়েছেন। (Three candidates were **appointed** at the one-day job fair.)

Event Type	Sub Type	Definition	Example
Socio-Economy	Disrupt	যখন অর্থনীতি, বাণিজ্য, শিল্প, উৎপাদন, সরবরাহ ব্যবস্থা, শিক্ষা, কর্মসংস্থান বা জনসেবাসংশ্লিষ্ট কোনো কার্যক্রম উল্লেখযোগ্যভাবে ব্যাহত, বিঘ্নিত বা স্থবির হয়। (When activities related to the economy, trade, industry, production, supply chains, education, employment, or public services are significantly disrupted, interrupted, or brought to a halt.)	আর্থিক সংকট বিশ্বব্যাপী সরবরাহ চেইনকে **ব্যাহত** করেছে। (The financial crisis has **disrupted** global supply chains.)
	Graduate	যখন কোনো ব্যক্তি বা আবেদনকারী কোনো শিক্ষাগত বা প্রতিযোগিতামূলক পরীক্ষায় উত্তীর্ণ হন, স্নাতক সম্পন্ন করেন বা আনুষ্ঠানিকভাবে শিক্ষাজীবনের একটি ধাপ সম্পন্ন করেন। (When a person or applicant passes an academic or competitive examination, graduates from an educational program, or formally completes a stage of education.)	এতে চূড়ান্তভাবে **উত্তীর্ণ** হয়েছেন ২৩,৯৮৫ জন। (A total of 23,985 candidates **passed** the examination.)
	Grow	যখন অর্থনীতি, শিল্প, বাণিজ্য বা অন্যান্য পরিমাপযোগ্য সূচকের প্রবৃদ্ধি, বৃদ্ধি বা সম্প্রসারণের কথা উল্লেখ থাকে। (When the growth, increase, or expansion of an economic indicator, industry, trade, or another measurable quantity is reported.)	বাংলাদেশ গত ১৩ বছরে গড়ে ৬.৬% জিডিপি **প্রবৃদ্ধি** অর্জন করেছে। (Bangladesh has achieved an average GDP **growth** of 6.6% over the past 13 years.)
	Import-Export	যখন কোনো পণ্য বিদেশ থেকে দেশে আমদানি করা হয় অথবা দেশ থেকে বিদেশে রপ্তানি করা হয়। (When goods are imported into a country from abroad or exported from a country to foreign markets.)	ভোজ্যতেলের চাহিদার একটি অংশ **আমদানি** করা বীজ থেকে তেল উৎপাদনের মাধ্যমে পূরণ করা হয়। (A portion of the demand for edible oil is met by producing oil from **imported** seeds.)
	Invest	যখন কোনো ব্যক্তি, প্রতিষ্ঠান বা রাষ্ট্র কোনো ব্যবসা, প্রকল্প, শিল্প বা দেশে অর্থ বা পুঁজি বিনিয়োগ করে। (When an individual, organization, or government invests money or capital in a business, project, industry, or country.)	থাইল্যান্ডের বিনিয়োগকারীদের বাংলাদেশে **বিনিয়োগে** আকৃষ্ট করা জরুরি। (It is important to attract Thai investors to **invest** in Bangladesh.)
	Recruit	যখন কোনো প্রতিষ্ঠান বা সংস্থা নির্দিষ্ট পদের জন্য নিয়োগ বিজ্ঞপ্তি প্রকাশ করে বা আবেদন আহ্বান করে। (When an organization or institution publishes a recruitment notice or invites applications for one or more positions.)	জনবল **নিয়োগের বিজ্ঞপ্তি** দিয়েছে বাংলাদেশ সেনাবাহিনী। (The Bangladesh Army has published a **recruitment notice**.)

Event Type	Sub Type	Definition	Example
	Trade	যখন সামগ্রিক বাণিজ্য কার্যক্রমের কথা সাধারণভাবে উল্লেখ থাকে। (When the text generally refers to overall trade activities.)	বড় ধরনের **বাণিজ্য** ঘাটতিতে পড়ে ক্রমাগত চলতি হিসাবের ঘাটতিও বেড়েছিল বাংলাদেশের। (Bangladesh's current account deficit continued to increase as the country experienced a large **trade** deficit.)
Technology	Launch-Service	যখন কোনো বিদ্যমান প্রতিষ্ঠান বা সংস্থা নতুন কোনো প্রযুক্তিপণ্য, ডিজিটাল সেবা, অ্যাপ্লিকেশন বা অনুরূপ সেবা আনুষ্ঠানিকভাবে চালু বা বাজারে উন্মোচন করে। (When an existing organization or institution officially launches or introduces a new technology product, digital service, application, or similar service.)	হজ অ্যাপ `লাব্বাইক' **চালু** করল মন্ত্রণালয়। (The ministry **launched** the `Labbaik' Hajj application.)
Transaction	Transfer-Money	যখন কোনো ব্যক্তি, প্রতিষ্ঠান বা সংস্থা কোনো পণ্য বা সেবা ক্রয়ের উদ্দেশ্য ব্যতীত অন্য ব্যক্তি বা প্রতিষ্ঠানের কাছে অর্থ প্রদান, গ্রহণ, ঋণ, অনুদান, দান বা আর্থিক সহায়তা প্রদান বা গ্রহণ করে। (When an individual, organization, or institution transfers, receives, lends, borrows, donates, or grants money to another party without the primary purpose of purchasing goods or services.)	বাণিজ্যিক ব্যাংকগুলো টাকা **ধার** করতে হলে আগের চেয়ে দশমিক ২৫ শতাংশ বেশি সুদ গুনতে হবে। (Commercial banks will have to pay 0.25 percentage points more in interest to **borrow** money.)
	Transfer-Ownership	যখন কোনো ব্যক্তি, প্রতিষ্ঠান, যানবাহন, অস্ত্র, সম্পদ বা অন্যান্য মালামালের মালিকানা ক্রয়, বিক্রয়, হস্তান্তর, দান বা ধার দেওয়া-নেওয়ার মাধ্যমে এক পক্ষ থেকে অন্য পক্ষের কাছে স্থানান্তরিত হয়। (When the ownership of a person-owned asset, organization, vehicle, weapon, property, or other goods is transferred from one party to another through purchase, sale, transfer, donation, or lending.)	সরকারের অনুমতি পেলে তারা আগে **ক্রয়কৃত** পেঁয়াজ বুধবার বাংলাদেশে পাঠাবেন। (Subject to government approval, they will ship the **purchased** onions to Bangladesh on Wednesday.)

CrisisKD: Five-Stage Knowledge Distillation for Aspect-Level Sentiment and Emotion Analysis in Crisis Discourse

Marko Haralović[1,2] **Onat Akca**[1] **Salih Eren Yücetürk**[1] **Minsi Li**[3] **Mariët Theune**[1]

[1]Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente
[2]Faculty of Electrical Engineering and Computing, University of Zagreb
[3]Faculty of Behavioural, Management and Social Sciences, University of Twente

Abstract

Identifying the target of emotional words or phrases in crisis situations, especially health-related ones, is important for understanding public concerns across cultural and linguistic contexts. We propose CrisisKD, a five-stage teacher–student knowledge distillation framework for aspect-level sentiment and emotion analysis on unannotated social media data. A teacher LLM generates aspect-level labels and reasoning traces that supervise a smaller student model across aspect extraction, syntactic parsing, opinion extraction, sentiment classification, and emotion classification. Using this framework, we construct and release a dataset containing 50,615 aspect-level labels, together with the annotation and fine-tuning scripts as open-source resources.[1] The resulting student supports end-to-end ABSA and emotion detection at substantially lower inference cost than the teacher. On a manually annotated 500-tweet gold set, the 5-task Qwen2.5-7B student improves over the untuned model by 7.9 F1 points on aspect extraction, 17.0 points on emotion accuracy, and 6.5 points on sentiment accuracy. On the external ABEA benchmark, CrisisKD improves the same-model Qwen2.5-7B ICL baseline by 2.8 F1 points on ATE and 3.8 F1 points on joint ATE+AEC.

Keywords: knowledge distillation; aspect-based sentiment and emotion classification; large language models; multi-step reasoning; crisis discourse; cross-domain generalization;

1 Introduction

During major health crises, such as the COVID-19 pandemic, public discourse becomes emotionally complex (Abd-Alrazaq et al., 2020). Understanding this discourse requires more than identifying whether a statement is positive or negative; we must also identify the specific topic, or "aspect," targeted by the emotion. For example, "fear" directed at the virus may imply willingness to cooperate with health measures, whereas "fear" directed at a vaccine may indicate reduced trust in its safety. This distinction is important because aggregate sentiment measures can obscure differences in public response (Pontiki et al., 2014). The same emotion may therefore signal different intentions depending on its target.

Aspect-Based Sentiment Analysis (ABSA) identifies topics and the sentiments expressed toward them, but standard approaches (Zhang et al., 2018) have largely focused on product reviews and coarse sentiment polarities rather than the fine-grained emotions, such as fear, anger, and hope, that shape behavior in crisis contexts. In this paper, we jointly model aspect extraction, aspect-based sentiment analysis, and aspect-level emotion detection on unannotated COVID-19 tweets.

A major barrier to aspect-level analysis is the lack of large COVID-19 datasets manually annotated for aspect-based sentiment and emotion classification (Banda et al., 2021). Existing datasets typically focus on named entities or sentence-level sentiment rather than aspect-level emotion categories. This limitation is particularly acute outside English, where annotated crisis data is even scarcer.

To reduce dependence on manual annotation, we propose CrisisKD, a five-stage teacher–student knowledge distillation framework. A large teacher model automatically annotates existing COVID-19 data and provides supervision for smaller student models. These students are more suitable for lower-resource deployment, and quantized variants can run on consumer hardware.

CrisisKD is informed by prior reasoning-based distillation work, including Syn-Chain (Fan et al., 2025), which shows that syntactic parsing and opinion extraction can improve ABSA knowledge distillation. Unlike approaches that assume existing aspect-level annotations, CrisisKD operates on raw

[1]https://github.com/MarkoHaralovic/CrisisKD

Proceedings of the 11th Workshop on Natural User-generated Text, pages 49–64
October 28, 2026 ©2026 Association for Computational Linguistics

text and generates both labels and reasoning traces for five tasks: aspect extraction, syntactic parsing, opinion extraction, sentiment classification, and emotion classification. The teacher outputs serve as supervision for the student, allowing it to learn both task labels and intermediate reasoning steps. Because the pipeline requires only raw text and a dependency parser, it is architecturally portable to other crisis domains and languages supported by suitable parsers such as spaCy (Honnibal et al., 2020), although such transfer remains to be empirically validated.

1.1 Research Questions and Contributions

Our work addresses three questions:

- **RQ1**: Can LLM-generated synthetic annotations enable effective ABSA training on domain-specific data without human aspect-level annotations?
- **RQ2**: Do auxiliary syntactic-parsing and opinion-extraction tasks improve student performance on aspect extraction, sentiment classification, and emotion classification?
- **RQ3**: How do aspect-level emotions and sentiments differ in COVID-19 discourse, and what insights do they provide?

Our main contributions are: (1) a multi-step reasoning framework with explicit aspect extraction and emotion classification, architecturally adaptable to other crisis domains and languages; (2) an openly released COVID-19 ABSA dataset with emotion labels, annotation code, teacher prompts, and fine-tuned model weights; and (3) a student model that supports both downstream crisis-discourse analysis and automatic annotation of unannotated data, evaluated on a manually annotated gold set of 500 tweets.

2 Related Work

Weak Supervision and Automatic Labeling Interpreting event-specific tweets, such as COVID-19 discourse, is challenging because manually annotated data are scarce. Researchers therefore use **weak supervision**, where noisy or automatically generated labels provide training signals. Automatically generated pseudo-labels have proved viable for aspect-based sentiment analysis, particularly when syntactic structure is incorporated (Negi et al., 2024; Ratner et al., 2020).

Syntactic Dependency in ABSA Identifying the precise target of a sentiment requires modeling the grammatical relations that connect opinions to their targets. Prior work shows that syntax helps disentangle complex sentiments in English and Chinese COVID-19 discourse (Hou et al., 2025). Accordingly, CrisisKD includes syntactic parsing as an auxiliary task.

Emotion Detection in Crisis Discourse Fine-grained emotional targets, such as fear and anxiety, are important for understanding public health compliance and response (Abadi et al., 2021). Previous work trained downstream models from lexicon-based emotion annotations (Salsabila et al., 2023), but fixed lexicons struggle with context-dependent phenomena such as sarcasm and implicit emotion. CrisisKD instead uses contextualized reasoning from a large language model.

Reasoning-Based Knowledge Distillation Recent work moves beyond label prediction by transferring reasoning traces from a teacher model to a smaller student model (Fan et al., 2025). Such traces provide useful inductive bias for ABSA and can outperform label-only supervision. Building on this broader direction, CrisisKD provides a complete five-stage pipeline for unannotated data, jointly covering aspect extraction, syntactic parsing, opinion extraction, sentiment classification, and emotion classification.

Instruction Tuning Prompt formulation also influences ABSA performance. PFInstruct (Cabello and Akujuobi, 2024) shows that NLP-specific task prefixes improve results across ABSA subtasks. We follow this principle in the design of the CrisisKD prompts.

Cross-Lingual and Cross-Domain Portability Many ABSA and emotion systems remain restricted to English (Wu et al., 2025; Zhang et al., 2023). Multilingual parsers such as spaCy (Honnibal et al., 2020) provide CoNLL-U representations for many languages, while multilingual COVID-19 datasets exist in Arabic, Spanish, and Chinese (Imran et al., 2022; Hou et al., 2025). Because CrisisKD requires only raw text, a dependency parser, and a multilingual teacher model such as Qwen2.5-32B (Yang et al., 2024), it is architecturally portable beyond English, although such transfer remains to be empirically validated.

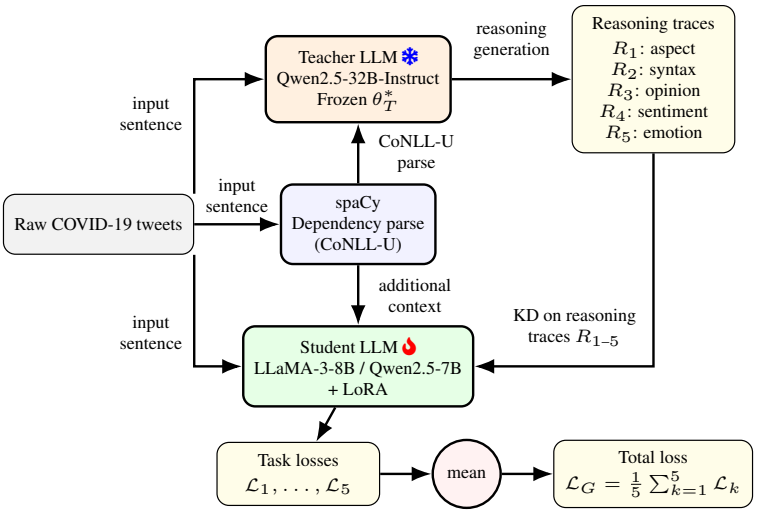


Figure 1: System overview of the proposed teacher–student framework. Each input sentence is parsed by spaCy into CoNLL-U format and, together with the raw sentence, is fed to a frozen teacher LLM (θ_T^*). The teacher produces multi-task reasoning traces R_1–R_5, which serve as knowledge-distillation targets for the student LLM (LLaMA-3-8B-Instruct or Qwen2.5-7B-Instruct, fine-tuned with LoRA). Task-specific generative losses $\mathcal{L}_1 \ldots \mathcal{L}_5$ are averaged into $\mathcal{L}_G$.

3 Proposed Method

3.1 Method Overview

To address the challenge of identifying emotional targets without large-scale manual annotation, we propose CrisisKD, a teacher–student knowledge distillation framework visualized in Fig. 1.

CrisisKD uses the reasoning capabilities of a large language model (Teacher) to synthesize training data containing explicit reasoning traces that explain each label. These traces are then used to fine-tune a smaller, efficient Student model. This allows the Student to learn the reasoning required to identify why a sentiment or emotion is directed at a specific target while retaining the computational efficiency needed for large-scale analysis.

3.2 Data Selection

We selected COVID-19 tweet datasets that provide complete text for pseudo-annotation and knowledge distillation. Among available resources, including TweetsCOV19 (Dimitrov et al., 2020), COVIDSenti (Naseem et al., 2021), COVID19 NLP (Miglani, 2020), TB-COV (Imran et al., 2022), and METS-CoV (Zhou et al., 2022), many release only tweet IDs and therefore require Twitter/X API access. We consequently use **COVIDSenti** (Naseem et al., 2021), with approximately 90,000 tweets labeled as positive, negative, or neutral, and **COVID19 NLP** (Miglani, 2020), with 48,000 tweets labeled using five sentence-level sentiment categories. Neither dataset provides aspect-level sentiment or emotion labels.

We therefore pseudo-annotate both datasets with the teacher model, generating five reasoning traces (R_{1-5}) per sentence. These synthetic annotations are then used for knowledge distillation and student training. For cross-dataset evaluation, we additionally use the manually annotated **ABEA** dataset (Zorenböhmer et al., 2026), which provides aspect and emotion annotations independent of our pseudo-annotation pipeline.

Preprocessing Tweets lacking personal opinion or emotion, including headline-style posts, links, and quoted news, were removed using rule-based filtering. The remaining tweets were normalized, stripped of emojis and URLs, and parsed with spaCy's (Honnibal et al., 2020) `en_core_web_sm` model into CoNLL-U format, with Unicode normalization performed using `ftfy` (Speer, 2019). Full filtering criteria and preprocessing details are provided in Appendix B.1.

3.3 Emotion Taxonomy

For aspect-level emotion classification, we adopt a 14-category emotion taxonomy designed for crisis discourse, together with one additional `no_emotion` fallback label for orphan annotations, failed parses, or cases where the teacher does not assign a valid emotion. The SenWave dataset (Yang et al., 2025) suggests the following seven categories: *optimistic*, *thankful*, *empathetic*,

pessimistic, *anxious*, *sad*, and *annoyed*. We extend this set with seven additional categories derived from fine-grained emotion taxonomies used in prior work (Demszky et al., 2020): *hopeful*, *proud*, *trustful*, *satisfied*, *scared*, *angry*, and *neutral*. We believe this taxonomy captures the details in emotional responses characteristic of pandemic discourse, distinguishing between related but distinct states such as *anxious* versus *scared*, or *optimistic* versus *hopeful*. The inclusion of no_emotion allows the model to handle factual statements about aspects where no affective content is present.

Unlike sentence-level emotion detection, our task assigns exactly one emotion label per aspect, enabling fine-grained analysis of how different targets within the same tweet evoke distinct emotional responses. For instance, a tweet may express *thankfulness* toward healthcare workers while simultaneously expressing *anxiousness* toward vaccine side effects.

This design reflects a social-psychological perspective on crisis emotions: distinct states such as *anxious* versus *scared*, or *optimistic* versus *hopeful*, map to different behavioral intentions and compliance patterns in public health contexts (Schröder et al., 2016). From the perspective of affect control theory, emotions are not merely internal states but socially and culturally shared sentiments that guide behavior (Schröder et al., 2016). Fine-grained aspect-level classification is therefore valuable beyond what coarse polarity models capture, particularly in multicultural crisis settings where the same emotional label may carry different behavioral implications across communities.

3.4 The Reasoning Pipeline

Our pipeline consists of five consecutive tasks.

Step 1: Aspect Extraction. Building on the syntactic structure, the model identifies aspects: specific entities, topics, or targets of opinion. For example, in "The vaccine rollout has been slow but the side effects are minimal," the aspects are "vaccine rollout" and "side effects." The teacher model generates a reasoning trace R_1 explaining why each span constitutes an aspect (e.g., "vaccine rollout" is identified because it represents a concrete policy action that opinions are expressed about). This task enables our student model to perform automatic aspect annotation.

Step 2: Syntactic Parsing. Dependency parses are obtained via spaCy (Honnibal et al., 2020) and converted to CoNLL-U format, a tabular representation that explicitly encodes each word's part-of-speech, head word, and dependency relation. This structured format is easier for LLMs to process than raw dependency trees (Fan et al., 2025; Matsuda et al., 2025). The teacher model generates a reasoning trace R_2 for syntactic parsing, learning to predict this structure from raw text. This serves two purposes: it provides syntactic features that inform the other tasks, and it acts as an auxiliary task that improves the student model's linguistic understanding.

Step 3: Opinion Extraction. For each identified aspect, the teacher model extracts the opinion terms expressing sentiment toward that aspect. In our example, "slow" expresses opinion toward "vaccine rollout" and "minimal" toward "side effects." Reasoning trace R_3 explains each opinion-aspect connection through dependency relationships, providing significant performance gains (Fan et al., 2025).

Step 4: Sentiment Classification. For each aspect-opinion pair, the model determines sentiment polarity (positive, negative, neutral). The reasoning trace R_4 explains why a particular sentiment label is appropriate given the opinion terms and broader context.

Step 5: Emotion Classification. For each aspect, the model identifies emotions expressed toward it (e.g., fear, anger, joy, trust, sadness). These categories are based on closed-set emotion taxonomies, which are relevant for crisis discourse. The reasoning trace R_5 explains the emotional content, which is more subtle than sentiment polarity.

Five reasoning traces were obtained from the teacher model Qwen2.5-32B-Instruct (Yang et al., 2024): R_1 for aspect extraction, R_2 for syntactic parsing, R_3 for opinion extraction, R_4 for sentiment classification, and R_5 for emotion extraction. The Qwen2.5-Instruct model family is reported to achieve state-of-the-art performance among open-weight models (Yang et al., 2024). Annotation of the corpus with Qwen2.5-32B-Instruct (Yang et al., 2024) on an NVIDIA L40 (48 GB) took approximately 7 days. The student models are Llama-3-8B (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024), selected based on their strong distillation performance (Fan et al., 2025). Both models are fine-tuned using a LoRA adapter (Hu et al., 2022) via knowledge distillation from the teacher model's reasoning traces. We

apply explicit teacher forcing between reasoning steps, i.e., step-level teacher annotations are provided as input rather than the student's predictions from previous step. We adopt LoRA as an effective substitute due to its efficiency and the computational cost of full fine-tuning (Hu et al., 2022).

Loss definition The unified training objective across all five tasks is formulated as a generative loss:

$$\mathcal{L}_G = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\log P\big(g_{i,t} \mid \hat{g}_{i,<t}, C_i\big), \quad (1)$$

where N denotes the number of training samples, T is the length of the output sequence for each sample, $g_{i,t}$ is the ground-truth token at position t for sample i, $\hat{g}_{i,<t}$ is the previously generated token sequence up to position $t-1$, and $P(g_{i,t} \mid \hat{g}_{i,<t}, C_i)$ is the conditional probability of generating $g_{i,t}$ given the context C_i (which encodes the input sentence, aspect, and task-specific prompt) and the prefix $\hat{g}_{i,<t}$ (Fan et al., 2025).

Golden Set Evaluation The final fine-tuned model is evaluated against 500 manually annotated tweets derived from the COVID-Senti (Naseem et al., 2021) and COVID19-NLP (Miglani, 2020) datasets. Annotations followed a structured coding protocol. Inter-annotator agreement was assessed using span-level F1 scores for aspect extraction, and accuracy for sentiment and emotion labels on aligned aspect spans; disagreements were resolved through consensus. The finalized annotations constitute the gold standard used for evaluating model performance.

The main methodological contribution of CrisisKD is a five-stage distillation pipeline that jointly generates aspect, syntax, opinion, sentiment, and emotion supervision from raw text. This enables the student to operate both as a downstream ABSA system and as an automatic annotator for previously unannotated data.

3.5 Dataset Statistics

Teacher annotations were aggregated over the COVID19NLP (Miglani, 2020) and COVID-Senti (Naseem et al., 2021) datasets. The resulting corpus contains 21,937 sentences and 50,615 aspect instances, corresponding to an average of 2.307 aspects per sentence. Neutral sentiment is the most frequent label (51.87%), followed by negative (36.13%) and positive sentiment (12.01%). The most frequent emotion labels are neutral (42.70%), annoyed (20.73%), and anxious (17.54%). Full label distributions and data-quality statistics are reported in Appendix B.2.

4 Experiments

We compare the performance of both student models on three tasks: aspect detection, sentiment classification, and emotion classification. Sentiment classification is a three-class problem (positive, negative, and neutral), while emotion classification uses 14 emotion categories and one additional `no_emotion` fallback label.

We validate the benefit of training on all five pipeline stages by removing components of the pipeline and evaluating the resulting model performance. This allows us to assess whether improvements in syntactic parsing or opinion extraction lead to improved downstream performance on the classification tasks.

We additionally compare model performance before and after fine-tuning to validate the efficacy of our approach. Teacher-model outputs are used as references, allowing us to measure alignment between teacher and student predictions. The model was trained with teacher forcing, whereas validation was performed end-to-end by propagating student responses from earlier tasks, thereby reproducing inference-time evaluation. We report task-level precision, recall, and macro F1, as well as accuracy for sentiment and emotion classification.

We evaluate CrisisKD in three settings: student–teacher fidelity, in-domain performance on a manually annotated gold set, and cross-dataset transfer on the external ABEA benchmark.

4.1 Training details

We use LoRA, a parameter-efficient fine-tuning method, where only a small subset of weights (adapter) is trained (Hu et al., 2022). For fair comparison between the two student models, Llama-3-8B-Instruct and Qwen2.5-7B-Instruct, we use the same fine-tuning recipe. Specifically, we use a learning rate of 2×10^{-5}, mixed-precision training, batch size of 8, and gradient accumulation of 8, resulting in an effective batch size of 64. We use an adapter rank of 16 targeting projection layers. For Qwen2.5-7B-Instruct we train 40M parameters ($\sim$0.5%), while for Llama-3-8B-Instruct we train 42M parameters ($\sim$0.5% of total parameters). The dataset is split into 80% training and 20% validation sets, and the exact splits are released with

Table 1: Student–teacher fidelity across training scenarios. We report precision, recall, F1, and accuracy for sentiment and emotion classification, and precision, recall, and F1 for aspect extraction for each model against teacher outputs. Bold values indicate the best performance for a model, and * denotes the best performance overall.

Model	Scenario	Aspect Extraction			Emotion Cls.				Sentiment Cls.			
		Prec.	Rec.	F1	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1	Acc.
Llama-3-8B-Instruct	No fine-tuning	0.341	0.515	0.411	0.566	0.635	0.515	0.733	0.876	0.909	0.891	0.911
	Fine-tune (3 tasks, w/o opinions and syntax)	**0.654**	0.643	**0.648**	0.699	0.656	0.669	0.850	**0.935***	**0.912***	**0.920***	**0.929**
	Fine-tune (4 tasks, w/o opinion extraction)	0.653	0.626	0.639	0.710	0.640	0.649	0.836	0.933	0.906	0.918	0.927
	Fine-tune (4 tasks, w/o syntactic parsing)	0.629	**0.646**	0.638	**0.724***	**0.657**	**0.675**	0.852	0.933	0.899	0.915	0.926
	Fine-tune (5 tasks)	0.649	0.622	0.635	0.698	0.640	0.638	**0.858**	0.933	0.906	0.918	0.927
Qwen2.5-7B-Instruct	No fine-tuning	0.466	0.461	0.463	0.561	**0.710***	0.587	0.791	0.903	0.905	0.903	0.922
	Fine-tune (3 tasks, w/o opinions and syntax)	0.638	0.658	0.648	0.699	0.682	0.670	0.852	0.930	0.900	0.915	**0.929***
	Fine-tune (4 tasks, w/o opinion extraction)	**0.696***	0.728	0.712	0.694	0.689	0.663	0.910	**0.933**	**0.901**	**0.916**	0.900
	Fine-tune (4 tasks, w/o syntactic parsing)	0.665	0.723	0.693	0.707	0.673	0.667	0.920	0.929	0.900	0.914	0.920
	Fine-tune (5 tasks)	0.681	**0.747***	**0.712***	**0.710**	0.696	**0.703***	**0.930***	0.923	0.901	0.912	0.910

the code. All student models were trained for 500 steps on a single NVIDIA L40 (48 GB), with training ranging 8-16 hours depending on the number of tasks in fine-tuning configuration.

4.2 Student–Teacher Fidelity Evaluation

Table 1 reports results for fine-tuning Llama-3-8B-Instruct and Qwen2.5-7B-Instruct. We evaluate four settings: (i) no fine-tuning, (ii) fine-tuning on downstream tasks only (3 tasks: aspect extraction, aspect-based sentiment classification, and emotion classification), (iii) fine-tuning on all five reasoning steps (adding syntactic parsing and opinion extraction), and (iv) ablations that omit one of the two additional reasoning tasks (syntax or opinion extraction) to test whether these traces improve student performance.

It is important to note that knowledge distillation in this setup is not fully model-agnostic: the token distribution and reasoning style are typically more similar within a model family than across families. Consequently, Qwen2.5-7B-Instruct may learn more effectively from the in-family teacher Qwen2.5-32B-Instruct than Llama-3-8B-Instruct.

Llama-3-8B-Instruct Results From Table 1, we draw the following conclusions for Llama-3-8B-Instruct: adding reasoning traces for syntactic parsing and opinion extraction does not provide meaningful benefit. The model achieves the best overall performance when fine-tuned only on the three downstream tasks (aspect extraction, sentiment classification, and emotion classification), reaching the highest aspect extraction F1 score of 64.8%, the highest sentiment classification accuracy of 92.9% across runs, and the second-highest emotion classification accuracy at 85.2%. Relative to the baseline, this corresponds to improvements of $+23.5$ pp in aspect extraction F1, $+1.8$ pp in sentiment classification accuracy, and $+11.7$ pp in emotion classification accuracy.

Qwen2.5-7B-Instruct Results For Qwen2.5-7B-Instruct, fine-tuning with reasoning traces yields substantial gains over the out-of-the-box baseline. The best overall configuration is obtained when fine-tuning on **all five tasks**, achieving the highest emotion classification accuracy (93.0%) and the strongest aspect extraction performance (F1 of 71.2%). Removing either syntactic parsing or opinion extraction slightly reduces overall performance, suggesting that the additional reasoning steps are beneficial for the Qwen student model. Relative to the baseline, the best model improves aspect extraction F1 by $+24.9$ pp, emotion classification accuracy by $+13.9$ pp, while sentiment classification, even though high 91.0%, drops by -1.2 pp compared to the baseline.

Student Results Comparison Comparing the best-performing configurations of both students, Llama (3 task fine-tuning) versus Qwen (5 task fine-tuning), we observe that the Qwen student consistently performs better on the downstream objectives. In particular, Qwen achieves higher as-

Table 2: Performance on the manually annotated gold set. We report accuracy and macro-F1 where applicable. Majority-class and uniform-random baselines are included for the classification tasks.

Task	Model	Acc.	F1
Aspect Extraction	Qwen2.5-7B Base	–	0.549
	Qwen2.5-7B FT (3)	–	0.583
	Qwen2.5-7B FT (5)	–	**0.628**
	Teacher: Qwen2.5-32B	–	***0.709***
Emotion Classification	Majority class	*0.551*	*0.051*
	Uniform random	*0.305*	*0.077*
	Qwen2.5-7B Base	0.508	0.569
	Qwen2.5-7B FT (3)	0.599	0.562
	Qwen2.5-7B FT (5)	**0.678**	**0.570**
	Teacher: Qwen2.5-32B	***0.710***	***0.599***
Sentiment Classification	Majority class	*0.537*	*0.233*
	Uniform random	*0.423*	*0.333*
	Qwen2.5-7B Base	0.658	0.692
	Qwen2.5-7B FT (3)	0.713	0.696
	Qwen2.5-7B FT (5)	**0.723**	**0.696**
	Teacher: Qwen2.5-32B	***0.755***	***0.713***

pect extraction performance (F1 score of 71.2% vs. 64.8%) and significantly higher emotion classification accuracy (93.0% vs. 85.2%), indicating better alignment with the teacher outputs when distilling within the same model family.

4.3 In-Domain Human Gold-Set Evaluation

Inter-coder agreement on the double-annotated subsets indicates generally consistent annotations across annotators. Aspect extraction achieves a mean F1 score of 73.0% across annotator pairs, while sentiment and emotion classification reach accuracies of 83.0% and 79.0%, respectively.

Table 2 reports results on the gold set of 500 tweets for the Qwen2.5-7B-Instruct baseline and our fine-tuned Qwen2.5-7B-Instruct students, trained on either 3 tasks or all 5 tasks in our pipeline, compared with the teacher model Qwen2.5-32B-Instruct. Fine-tuned models, as expected, outperform the baseline, with full fine-tuning across all 5 tasks showing better performance than 3-task fine-tuning. The performance of the fine-tuned model narrows the gap to the teacher model, demonstrating the effectiveness of our knowledge distillation framework.

4.4 Cross-Dataset ABEA Evaluation

Table 3 evaluates cross-dataset transfer to the ABEA benchmark. Compared with the same Qwen2.5-7B model used with in-context learning (ICL) our fine-tuned model improves ATE F1 from 0.385 to 0.413 and joint ATE+AEC F1 from 0.228 to 0.266. This indicates that our framework enables the 7B model to recover more relevant aspect and aspect–emotion instances than prompting the same model with ICL alone, narrowing the gap to the larger Qwen3-14B model.

Table 3: Performance on the ABEA test set. ATE denotes Aspect Term Extraction, AEC denotes Aspect Emotion Classification, and ICL denotes in-context learning. We compare our Qwen2.5-7B model against a same-size ICL baseline and a larger Qwen3-14B ICL model as an upper-bound reference.

Task	Model	Rec.	Prec.	F1
ATE	Qwen3-14B ICL	0.468	0.427	**0.447**
	Qwen2.5-7B ICL baseline	0.303	0.529	0.385
	Ours: Qwen2.5-7B	**0.359**	0.486	**0.413**
ATE + AEC	Qwen3-14B ICL	0.358	0.327	**0.342**
	Qwen2.5-7B ICL baseline	0.179	0.313	0.228
	Ours: Qwen2.5-7B	**0.231**	**0.313**	**0.266**

4.5 Emotion–Aspect Associations via Enrichment Analysis

To better characterize fine-grained emotional structure beyond aggregate distributions, we analyze the association between aspects and emotions using a log-enrichment metric (Table 4). This measure points out aspects for which a given emotion is disproportionately expressed relative to its overall frequency in the dataset.

The results reveal that different aspects exhibit distinctive emotional signatures. Positive, socially oriented emotions (e.g., *proud*, *trustful*, *empathetic*) are strongly associated with human-centered aspects such as *doctor* and *people*, reflecting appreciation and collective solidarity. In contrast, negative emotions are concentrated around structurally different aspects: *angry* and *annoyed* are primarily linked to political and media-related aspects (e.g., *racism*, *trump*, *media*), while *pessimistic* and *anxious* are associated with broader systemic concerns such as *economy*, *pandemic*, and *travel*. Importantly, the same aspect appears across multiple emotion categories with different strengths. For example, *doctor* is strongly enriched for *proud*, *trustful*, and *empathetic*, while *shopping* is associated with both *optimistic* and *satisfied*. This shows aspect-level sentiment cannot be fully captured by coarse polarity labels, as distinct emotional inter-

Emotion	Top aspects (log-enrichment)
Proud	*doctor* (4.04), *people* (1.69), *vaccine* (1.48), *quarantine* (1.47), *shopping* (1.45)
Trustful	*doctor* (3.24), *hand sanitizer* (2.53), *World Health Organization* (2.52), *mask* (2.38), *government response* (1.89)
Optimistic	*shopping* (3.14), *health* (1.75), *cure* (1.74), *food* (1.68), *price* (1.66)
Thankful	*doctor* (3.13), *health* (2.48), *news* (2.26), *grocery/supermarket/store* (1.98), *shopping* (1.82)
Empathetic	*doctor* (3.06), *people* (2.43), *food* (2.15), *health* (2.10), *shopping* (1.78)
Hopeful	*cure* (3.04), *vaccine* (2.48), *news* (1.48), *shopping* (1.30), *world* (1.08)
Pessimistic	*economy* (3.03), *world* (2.69), *travel* (1.30), *italy* (1.15), *outbreak* (1.00)
Satisfied	*grocers response* (2.81), *price* (2.52), *shopping* (2.28), *food* (2.01), *flights* (1.73)
Sad	*doctor* (2.57), *deaths* (1.77), *quarantine* (1.19), *people* (1.12), *news* (0.92)
Angry	*racism* (2.50), *trump* (1.93), *public response* (1.84), *government response* (1.63), *government* (1.61)
No emotion	*vaccine* (2.39), *mask* (2.16), *doctor* (2.06), *shopping* (1.67), *government response* (0.80)
Annoyed	*media* (1.59), *panic buying* (1.33), *racism* (1.26), *trump* (1.24), *price* (1.03)
Anxious	*spread* (0.89), *travel* (0.83), *pandemic* (0.70), *economy* (0.65), *food* (0.40)
Neutral	*cruise ship* (0.58), *death toll* (0.58), *sars* (0.55), *wuhan* (0.55), *flights* (0.50)
Scared	*virus* (0.52), *covid/coronavirus* (0.26), *symptoms* (0.04), *panic* (-0.10), *travel* (-0.15)

Table 4: Top aspects per emotion ranked by log-enrichment, defined as $\log(P(\text{emotion} \mid \text{aspect})/P(\text{emotion}))$, over the top 50 aspects (minimum support $n \geq 200$). Positive values indicate that the emotion is more common for that aspect than in the dataset overall, negative values indicate underrepresentation, and 0 indicates baseline-level prevalence. Rows are ordered by the strongest top-ranked log-enrichment value in each emotion category.

pretations coexist within the same topic.

5 Discussion

With respect to the research questions introduced in Section 1.1, we organize the following sections to show how our findings support each question.

5.1 Effectiveness of Knowledge Distillation

Our results show that student models achieve substantial improvements across aspect extraction, sentiment and emotion classification. For Llama-3-8B-Instruct, the best configuration scores a $+23.5$ pp improvement in aspect extraction F1, a $+1.8$ pp improvement in sentiment classification accuracy, and a $+11.7$ pp improvement in emotion classification accuracy over the non-fine-tuned baseline. Similarly, Qwen2.5-7B-Instruct achieves large gains in aspect extraction ($+24.9$ pp F1) and emotion classification ($+13.9$ pp accuracy), providing evidence for the effectiveness of knowledge distillation with reasoning traces.

To assess the contribution of syntactic parsing and opinion extraction, we train models using all five tasks and ablate omitting one or both of these auxiliary tasks. For Llama-3-8B-Instruct, adding syntactic parsing and opinion extraction does not lead to additional gains. We attribute this to cross-architecture distribution mismatch, where reasoning traces generated by Qwen2.5-32B-Instruct may not align well with the internal representations of Llama-3-8B-Instruct, or where limited training capacity is spent modeling the teacher reasoning style rather than improving downstream performance.

In contrast, for the Qwen2.5-7B-Instruct student model, both syntactic parsing and opinion extraction provide measurable benefits. Fine-tuning on all five tasks results in the best overall performance, suggesting that the additional reasoning tasks are beneficial when teacher and student models belong to the same model family.

Overall, the results support claim that the proposed reasoning pipeline is effective. Exploring cross-architecture pairings and longer training schedules remains for future work.

5.2 Emotion Patterns in COVID-19 Discourse

Based on the teacher annotations, we observe that the majority of aspect-level sentiments are neutral (51.87%), and that 42.70% of aspect-level emotion labels are also neutral. At the sentence level, where annotations are available, we aimed to maintain similar label distributions.

It is possible for a sentence to exhibit an overall positive or negative valence while only the main aspect is assigned a corresponding sentiment/emotion label. Other, less central aspects that function as auxiliary nouns may be labeled as neutral because no explicit sentiment or emotional valence is expressed toward them. As a result, the distribution of neutral sentiment and emotion labels is skewed.

Another observation is the prevalence of negative sentiment labels, which occur approximately

three times more frequently than positive labels, and negatively associated emotions, which appear about five times more often than positive ones. This pattern reflects the general negativity present in pandemic discourse (Abadi et al., 2021; Lamsal, 2021). In crisis situations, such as COVID-19 pandemic, it is therefore particularly important to identify the specific targets of negative emotions, which highlights the value of our aspect-level analysis.

6 Conclusion

We propose CrisisKD, a five-stage teacher–student knowledge distillation pipeline for aspect-level sentiment and emotion analysis on unannotated COVID-19 tweets. The pipeline combines aspect extraction, syntactic parsing, opinion extraction, sentiment classification, and emotion classification, using a 14-category emotion taxonomy with an additional `no_emotion` fallback label. With Qwen2.5-32B as the teacher, we annotate 21,937 tweets with 50,615 aspect-level labels and fine-tune smaller student models, Llama-3-8B-Instruct and Qwen2.5-7B-Instruct, using LoRA. We evaluate the resulting models against both a manually annotated gold set of 500 tweets and the independently annotated ABEA benchmark. CrisisKD requires only raw text and a dependency parser, making it architecturally portable to other languages and crisis domains. Our contributions are:

- We propose CrisisKD, a five-stage distillation pipeline for aspect-level sentiment and emotion analysis on unannotated data.
- We construct and release a COVID-19 ABSA dataset with 50,615 annotated aspects across 21,937 tweets, together with annotation code and fine-tuned model weights.
- We demonstrate effectiveness on both human-annotated and external benchmarks. On a manually annotated 500-tweet gold set, the 5-task Qwen2.5-7B student improves over the untuned model by 7.9 F1 points in aspect extraction, 17.0 points in emotion accuracy, and 6.5 points in sentiment accuracy. On the external ABEA benchmark, it improves over the same-model ICL baseline by 2.8 F1 points on ATE and 3.8 F1 points on joint ATE+AEC.

Limitations

The framework relies on teacher-generated annotations, so teacher errors and biases may propagate to the student models. Although the teacher generally produced well-formed outputs, its annotations should be treated as noisy supervision rather than ground truth. In particular, it occasionally generated semantically overlapping aspect spans; because we prioritized recall over precision, we did not apply additional aspect-merging or span-refinement beyond prompt tuning. The 14-category emotion taxonomy may also miss subtle or culturally dependent expressions, while annotation with Qwen2.5-32B remains computationally expensive.

Nevertheless, the distilled students support lower-cost inference, including deployment on consumer hardware through quantization. The pipeline can reduce the cost of constructing domain-specific ABSA resources, and its reasoning traces support error analysis and system refinement. Improved prompting, confidence-based filtering, or post-processing may further improve annotation precision. Although the method is architecturally portable to other domains and languages with suitable parsers, such transfer remains to be empirically validated.

Ethical Considerations

This work studies aspect-based sentiment and emotion analysis on publicly available COVID-19 Twitter datasets. The datasets used in this research, COVIDSenti (Naseem et al., 2021) and COVID19NLP (Miglani, 2020), consist of social media posts released for research purposes. No new human participant data were collected. Data was pseudo- and manually annotated, and annotations are released to support future research. The datasets do not contain direct personally identifiable information, reducing privacy risks, and the analysis focuses on collective discourse patterns rather than individual users.

However, analyzing emotional content in social media raises ethical concerns. Systems that identify sentiment or emotion may be misused for surveillance, monitoring without consent, or manipulative targeting. While this work aims to support crisis communication and public opinion analysis, similar methods could negatively affect privacy or autonomy. Practitioners should carefully evaluate such risks.

The gold standard dataset was annotated by four volunteer annotators from the research team, without financial compensation. Annotators were trained using a coding scheme defining emotion

and sentiment categories, examples, and rules for aspect spans, sarcasm, and edge cases. A calibration session was conducted before independent annotation. As annotators were team members performing an analytical task, this work did not undergo IRB review. No personal annotator data was collected, and participation was voluntary.

Bias and generalizability remain concerns. Social media data may not represent the full diversity of affected populations, and Twitter users are not demographically representative. Cultural differences also influence emotional expression. Additionally, the pseudo-annotation process relies on a large language model, which may introduce biases or errors. Thus, annotations should be interpreted as approximations of individuals' internal emotional states.

Although the proposed framework can be adapted to other languages and domains when raw text and linguistic analysis are available, the present study evaluates the approach only on COVID-19 Twitter data. The method assumes that aspect, sentiment, and emotion relations can be inferred from short-form text and syntactic structure; performance may degrade when these assumptions are violated (e.g., sarcasm, implicit context, or culturally dependent expressions). As a result, transfer to new settings remains to be validated with domain-specific data.

Finally, training large models has environmental implications. To mitigate this, the framework uses knowledge distillation and parameter-efficient fine-tuning (LoRA), reducing computational requirements, with only 0.5% of model parameters trained.

To support transparency and reproducibility, we aim to open-source training and evaluation code, along with derived annotation artifacts where licensing permits, and future work should independently assess dataset and method bias to ensure ethical use of the proposed approach.

Acknowledgments

Computational resources were provided by the University of Twente High-Performance Computing infrastructure. Travel grants were provided by the Faculty of Electrical Engineering and Computing at the University of Zagreb.

References

David Abadi, Irene Arnaldo, and Agneta Fischer. 2021. Anxious and angry: Emotional responses to the covid-19 threat. *Frontiers in Psychology*, 12:676116.

Alaa Abd-Alrazaq, Dari Alhuwail, Mowafa Househ, Mounir Hamdi, and Zubair Shah. 2020. Top concerns of tweeters during the COVID-19 pandemic: Infoveillance study. *Journal of Medical Internet Research*, 22(4):e19016.

Juan M. Banda, Ramya Tekumalla, Guanyu Wang, Jingyuan Yu, Tuo Liu, Yuning Ding, Ekaterina Artemova, Elena Tutubalina, and Gerardo Chowell. 2021. A large-scale COVID-19 Twitter chatter dataset for open scientific research—an international collaboration. *Epidemiologia*, 2(3):315–324.

Laura Cabello and Uchenna Akujuobi. 2024. It is simple sometimes: A study on improving aspect-based sentiment analysis performance. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6597–6610, Bangkok, Thailand. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Dimitar Dimitrov, Erdal Baran, Pavlos Fafalios, Ran Yu, Xiaofei Zhu, Matthäus Zloch, and Stefan Dietze. 2020. TweetsCOV19 – a knowledge base of semantically annotated tweets about the COVID-19 pandemic. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2991–2998. Association for Computing Machinery.

Rui Fan, Shu Li, Tingting He, and Yu Liu. 2025. Aspect-based sentiment analysis with syntax-opinion-sentiment reasoning chain. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3123–3137, Abu Dhabi, UAE. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. 2024. The Llama 3 herd of models.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python. Zenodo.

Linlin Hou, Wenhui Tu, Ting Yu, Ting Jiang, Mohamed Bah, Zenghui Xu, Yu Zhang, Gaoming Yang, and Ji Zhang. 2025. Aspect-based sentiment analysis for covid-19: A heterogeneous graph convolutional network approach. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 24(6):55:1–55:26.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Muhammad Imran, Umair Qazi, and Ferda Ofli. 2022. TBCOV: Two billion multilingual COVID-19 tweets with sentiment, entity, geo, and gender labels. *Data*, 7(1):8.

Rabindra Lamsal. 2021. Design and analysis of a large-scale COVID-19 tweets dataset. *Applied Intelligence*, 51(5):2790–2804.

Hiroshi Matsuda, Chunpeng Ma, and Masayuki Asahara. 2025. Step-by-step instructions and a simple tabular output format improve the dependency parsing accuracy of LLMs. In *Proceedings of the 18th International Conference on Parsing Technologies (IWPT, SyntaxFest 2025)*, pages 11–19, Ljubljana, Slovenia. Association for Computational Linguistics.

Aman Miglani. 2020. Coronavirus tweets nlp - text classification. Kaggle dataset. Accessed 2026-01-22.

Usman Naseem, Imran Razzak, Matloob Khushi, Peter W. Eklund, and Jinman Kim. 2021. COVIDSenti: A large-scale benchmark twitter data set for COVID-19 sentiment analysis. *IEEE Transactions on Computational Social Systems*, 8(4):1003–1015.

Gaurav Negi, Rajdeep Sarkar, Omnia Zayed, and Paul Buitelaar. 2024. A hybrid approach to aspect based sentiment analysis using transfer learning. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 647–658, Torino, Italia. ELRA and ICCL.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2020. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2–3):709–730.

Salsabila, Salsabila Mazya Permataning Tyas, Yasinta Romadhona, and Diana Purwitasari. 2023. Aspect-based sentiment and correlation-based emotion detection on tweets for understanding public opinion of covid-19. *Journal of Information Systems Engineering and Business Intelligence*, 9(1):84–94.

Tobias Schröder, Jesse Hoey, and Kimberly B. Rogers. 2016. Modeling dynamic identities and uncertainty in social interactions: Bayesian affect control theory. *American Sociological Review*, 81(4):828–855.

Robyn Speer. 2019. ftfy. Zenodo. Version 5.5.1.

ChengYan Wu, Bolei Ma, Yihong Liu, Zheyu Zhang, Ningyuan Deng, Yanshu Li, Baolan Chen, Yi Zhang, Yun Xue, and Barbara Plank. 2025. M-ABSA: A multilingual dataset for aspect-based sentiment analysis. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2530–2557, Suzhou, China. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report.

Qiang Yang, Xiuying Chen, Changsheng Ma, Rui Yin, Xin Gao, and Xiangliang Zhang. 2025. Senwave: A fine-grained multi-language sentiment analysis dataset sourced from covid-19 tweets.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2023. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*, 35(11):11019–11038.

Peilin Zhou, Zeqiang Wang, Dading Chong, Zhijiang Guo, Yining Hua, Zichang Su, Zhiyang Teng, Jiageng Wu, and Jie Yang. 2022. METS-CoV: A dataset of medical entity and targeted sentiment on COVID-19 related tweets. In *Advances in Neural Information Processing Systems*, volume 35, pages 21916–21932. Curran Associates, Inc.

Christina Zorenböhmer, Sebastian Schmidt, David Hanny, and Bernd Resch. 2026. EmoGRACE: aspect-based emotion analysis for social media data. *Social Network Analysis and Mining*, 16:42.

Appendix

A Task Logic and Example Annotations

A.1 Task Logic Visualization

Figure 2 presents the reasoning structure of the proposed framework.

Given a raw tweet, the model proceeds through five consecutive tasks: aspect extraction, syntactic parsing, opinion extraction, sentiment classification, and emotion classification. This decomposition follows prior work in aspect-based sentiment analysis and structured reasoning (Pontiki et al., 2014; Zhang et al., 2023; Fan et al., 2025), enabling the model to identify both targets and their associated sentiment and emotion.

The visualization emphasizes pipeline hierarchy. Aspect extraction determines the opinion targets, syntactic parsing provides grammatical structure, and opinion extraction links expressions to aspects. These intermediate representations then support aspect-level sentiment and emotion prediction. In our teacher–student setting, the teacher produces both labels and reasoning traces for each step, and the student is trained to imitate this full process, similar to recent structured reasoning approaches in ABSA (Fan et al., 2025).

A.2 Example Annotations

To illustrate the complete reasoning pipeline, we present representative examples that demonstrate our model's capability to handle diverse COVID-19 discourse. These examples showcase the five-task reasoning chain (R1–R5) and highlight how the model disambiguates sentiment and emotion at the aspect level.

A.2.1 Handling Context

Figure 3 presents two contrasting cases that demonstrate the importance of aspect-level analysis.

The first example illustrates the model's ability to correctly identify metaphorical usage of "Coronavirus" (referring to Corona beer rather than the virus), resulting in neutral sentiment and satisfied emotion despite surface-level ambiguity. The syntactic parsing (R2) and opinion extraction (R3) steps are crucial here, as they reveal the humorous context through dependency relationships.

In contrast, the second example shows genuine health concern, where incomplete grammar and informal language ("wish coronavirus not that serious") are correctly interpreted as expressing negative sentiment and anxiety toward the pandemic situation. These examples demonstrate how the reasoning traces enable the model to navigate linguistic complexity and contextual meaning that would challenge traditional sentiment analysis approaches.

A.2.2 Handling Different Emotions in Multi-Aspect Example

Figure 4 demonstrates the model's capability on a more complex tweet containing two distinct evaluative targets.

The model successfully identifies both "media censorship" and "coronavirus outbreak" as separate aspects, performing independent reasoning chains for each. While both aspects receive negative sentiment classifications (R4), the emotion labels (R5) differ: anger toward censorship policies versus fear regarding outbreak severity. This distinction is semantically meaningful since anger reflects frustration with human decisions (censorship), while fear responds to the health threat itself.

The ability to extract and independently analyze multiple aspects within a single tweet is essential for understanding nuanced public discourse during crisis situations, where opinions often target multiple entities simultaneously with varying emotional valences.

A.2.3 Summary

These examples collectively demonstrate our model's capabilities: handling informal social media language, disambiguating metaphorical and literal usage, attributing sentiments and identifying fine-grained emotions to individual aspects.

B Emotional and Sentiment Analysis of Teacher Annotations

B.1 Dataset Preprocessing

News Filtering A challenge using these datasets for aspect-based sentiment analysis is the number of "news headline" style tweets, such as factual statements, news links, and quoted headlines that lack personal opinion or emotion, and therefore introduce irrelevant noise during training. Such tweets are unsuitable for ABSA, as they do not express aspect-level sentiment. Tweets identified as news were removed during preprocessing. News tweets were detected using patterns such as URLs, common headline-style prefixes, source attributions, and truncated headline snippets.

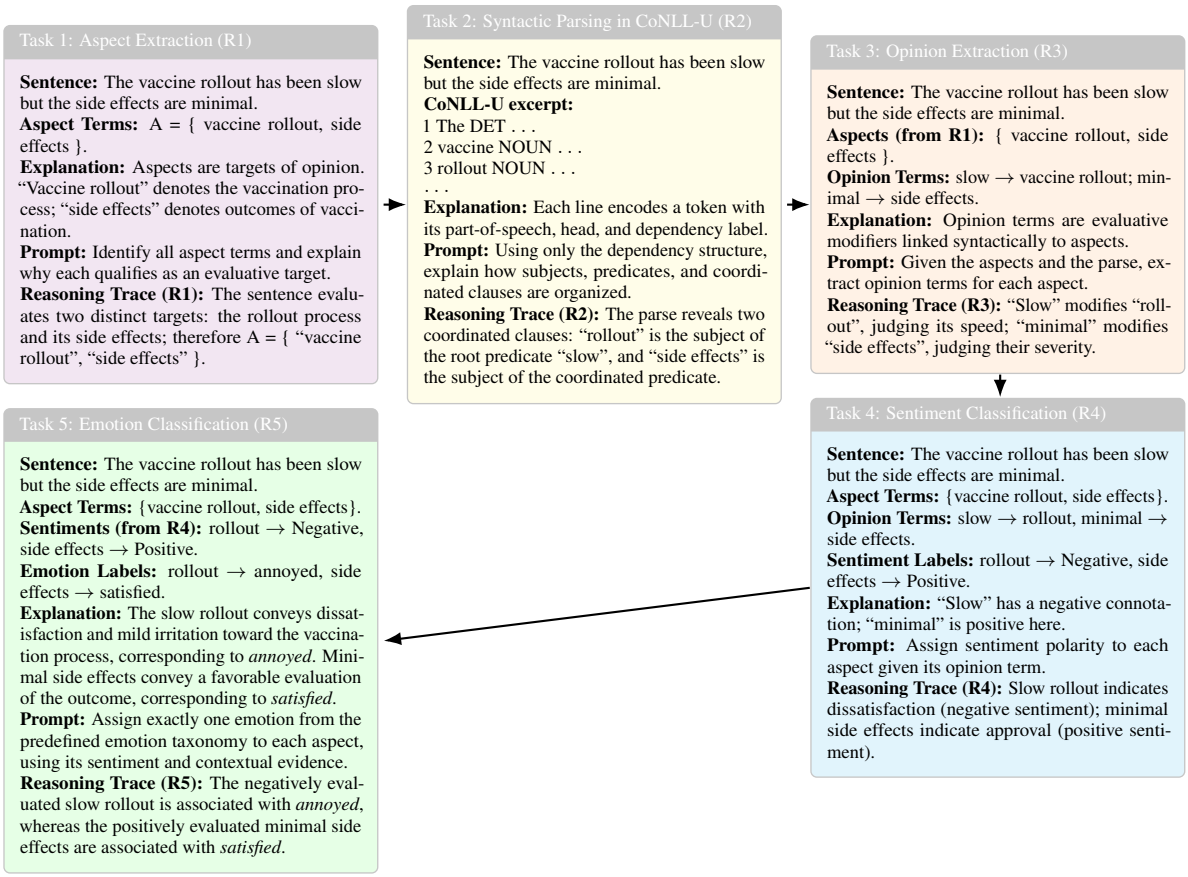


Figure 2: Reasoning example for aspect-based sentiment analysis and emotion detection on a single sentence via our teaching framework.

Example 1: Metaphorical Usage

INPUT TWEET:
"Hell, I have a 12 pack of Coronavirus in my fridge. It only makes you sick if you drink the whole thing."

REASONING_CHAIN_OUTPUT:
[R1] ASPECT:
"Coronavirus" (Metaphor)
→ Trace: "Refers to product/beer, not virus."
[R2] SYNTAX:
"Coronavirus" detected as proper noun modifying "pack" (compound noun phrase).
[R3] OPINION:
Trace: "Light-hearted joke... dismissive attitude."
[R4] SENTIMENT:
NEUTRAL (Context is humor, not health opinion).
[R5] EMOTION CLASS:
SATISFIED (Trace: "The playful tone conveys a positive, contented attitude.").

Example 2: Genuine Health Concern

INPUT TWEET:
"wish coronavirus not that serious and the effected ppl number stop increase."

REASONING_CHAIN_OUTPUT:
[R1] ASPECT:
"coronavirus" (detected as key aspect)
[R2] SYNTAX:
"coronavirus" functions as the main subject.
[R3] OPINION:
Trace: "The speaker expresses a desire for the coronavirus situation to be less severe..."
[R4] SENTIMENT:
NEGATIVE (The sentiment... is negative. Because the speaker expresses a wish for the situation to be less severe...)
[R5] EMOTION CLASS:
ANXIOUS ("The speaker... affected individuals to stop...")

Figure 3: Example reasoning traces for two COVID-19 tweets demonstrating the complete annotation pipeline. Left: sarcastic reference requiring contextual sentiment disambiguation. Right: anxious expression of genuine health concern with negative sentiment.

Text Preprocessing The remaining tweets underwent additional preprocessing, including emoji removal, Unicode normalization using the `ftfy` (Speer, 2019) library, and URL stripping. We then performed syntactic parsing using spaCy's (Honnibal et al., 2020) `en_core_web_sm` model, converting each tweet to CoNLL-U format, which encodes token-level information such as lemma, part-of-speech tags, and dependency relations.

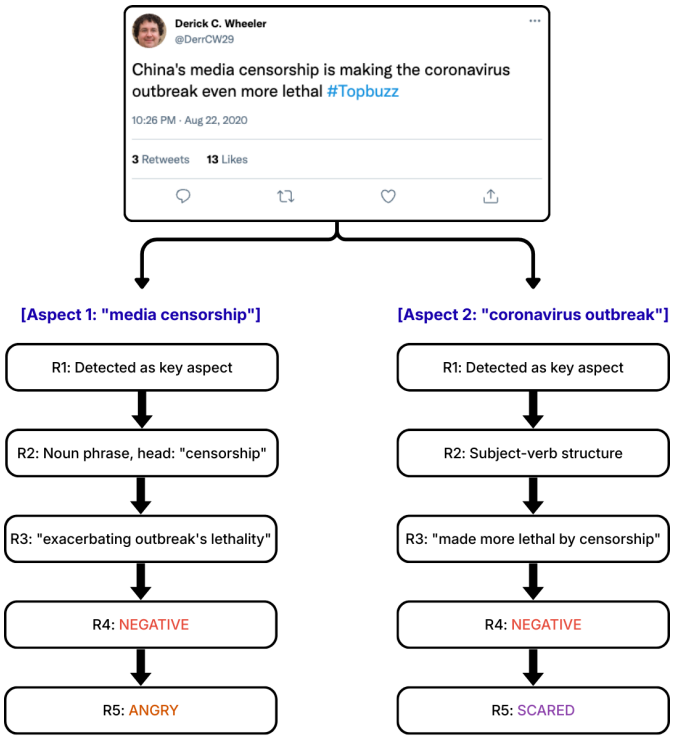


Figure 4: Illustrative reconstruction of a real tweet. Display name, profile image, username, timestamp, and engagement metadata are fictionalized. Reasoning trace across multiple aspects for a COVID-19 tweet. The model identifies two distinct aspects and performs the complete annotation pipeline for each, resulting in shared negative sentiment but divergent emotions: *angry* toward media censorship versus *scared* regarding the outbreak itself.

Table 5: Corpus summary based on Qwen2.5-32B annotations.

Metric	Value
Total sentences	21,937
Total aspects	50,615
Average aspects per sentence	2.307

Table 6: Aspect-level sentiment distribution (N = 50,615).

Sentiment	Count	Percent
Neutral	26,249	51.87%
Negative	18,288	36.13%
Positive	6,078	12.01%

B.2 Dataset Statistics

We report statistics computed from the Qwen2.5-32B (Yang et al., 2024) teacher annotations, aggregated over the COVID19NLP (Miglani, 2020) and COVIDSenti (Naseem et al., 2021) datasets.

The resulting corpus contains 21,937 sentences and 50,615 aspect instances, corresponding to an average of 2.307 aspects per sentence. During aggregation, we detected 13 orphan sentiment labels and 13 orphan emotion labels, defined as entries with missing aspect identifiers in the teacher model's JSON output. These entries correspond to 0.026% of the aspect instances and were excluded from the training data and all subsequent analyses.

We analyze the distribution of emotion annotations across aspect categories to better understand the structure of public discourse captured by our teacher model.

To improve interpretability, semantically equivalent aspect mentions (e.g., *coronavirus*, *covid*, *corona pandemic*, *covid crisis*) are normalized into unified aspect categories.

Figure 5 presents the emotion distributions for the 50 most frequent aspects. The results show that aspects related to pandemic are primarily associated with negative emotions, particularly anxiety and annoyance. In contrast, consumer-related aspects (e.g., *shopping*, *food*) exhibit more diverse emotional profiles.

The *vaccine* aspect is associated with hopeful emotion, reflecting expectations of recovery. In contrast, aspects related to political entities (e.g., *Trump*, *government response*) are largely associated with annoyance, indicating frustration toward

Table 7: Aspect-level emotion distribution ($N = 50{,}615$). The *no emotion* label indicates that the teacher did not assign an emotion to the aspect.

Emotion	Count	Percent
Neutral	21,608	42.70%
Annoyed	10,488	20.73%
Anxious	8,877	17.54%
Hopeful	1,640	3.24%
Scared	1,552	3.07%
Empathetic	1,436	2.84%
Satisfied	1,017	2.01%
Angry	1,014	2.00%
Optimistic	818	1.62%
Pessimistic	684	1.35%
Trustful	550	1.09%
Sad	449	0.89%
Thankful	262	0.52%
Proud	150	0.30%
No emotion	70	0.14%

decision-making and policy handling.

Overall, negative emotional valence dominates the dataset, with anxiety and annoyance being the most prevalent emotions, as public discourse during the pandemic is characterized by uncertainty and frustration. Aspects related to the pandemic often evoke anxiety due to perceived health risks and uncertainty, while secondary effects such as price increases, supply shortages, and quarantine measures contribute to annoyance and anger. Due to the large number of neutral tweets, where users write summaries or reports regarding certain aspects, neutral emotions account for a substantial proportion of each aspect's emotion distribution.

C Human Annotation Set

The gold standard dataset was annotated by four volunteer annotators from the research team, without financial compensation. Annotators were trained using a coding scheme defining emotion and sentiment categories, examples, and rules for aspect spans, sarcasm, and edge cases. A calibration session was conducted before independent annotation.

In total, annotators labeled 500 tweets with intentionally overlapping assignments to enable agreement analysis. Annotator 1 labeled tweets 1–250, Annotator 2 labeled tweets 250–500, Annotator 3 labeled tweets 1–125 and 375–500, and Annotator 4 labeled tweets 125–375.

This design ensures that each subset of tweets is annotated by at least two annotators, allowing us to examine inter-coding agreement, following standard practices in dataset construction for sentiment and emotion analysis (Demszky et al., 2020; Naseem et al., 2021).

Pair	Aspect (F1)	Sent.	Emo.
A1–A3	0.7493	0.8635	0.8229
A1–A4	0.7147	0.8289	0.7909
A2–A4	0.7386	0.7676	0.7042
A2–A3	0.7101	0.8516	0.8438
Mean	0.7282	0.8279	0.7905

Table 8: Inter-annotator agreement (aspect F1, sentiment accuracy, and emotion agreement).

We report aspect agreement in F1 score, sentiment accuracy, and emotion accuracy between overlapping annotator pairs in Table 8.

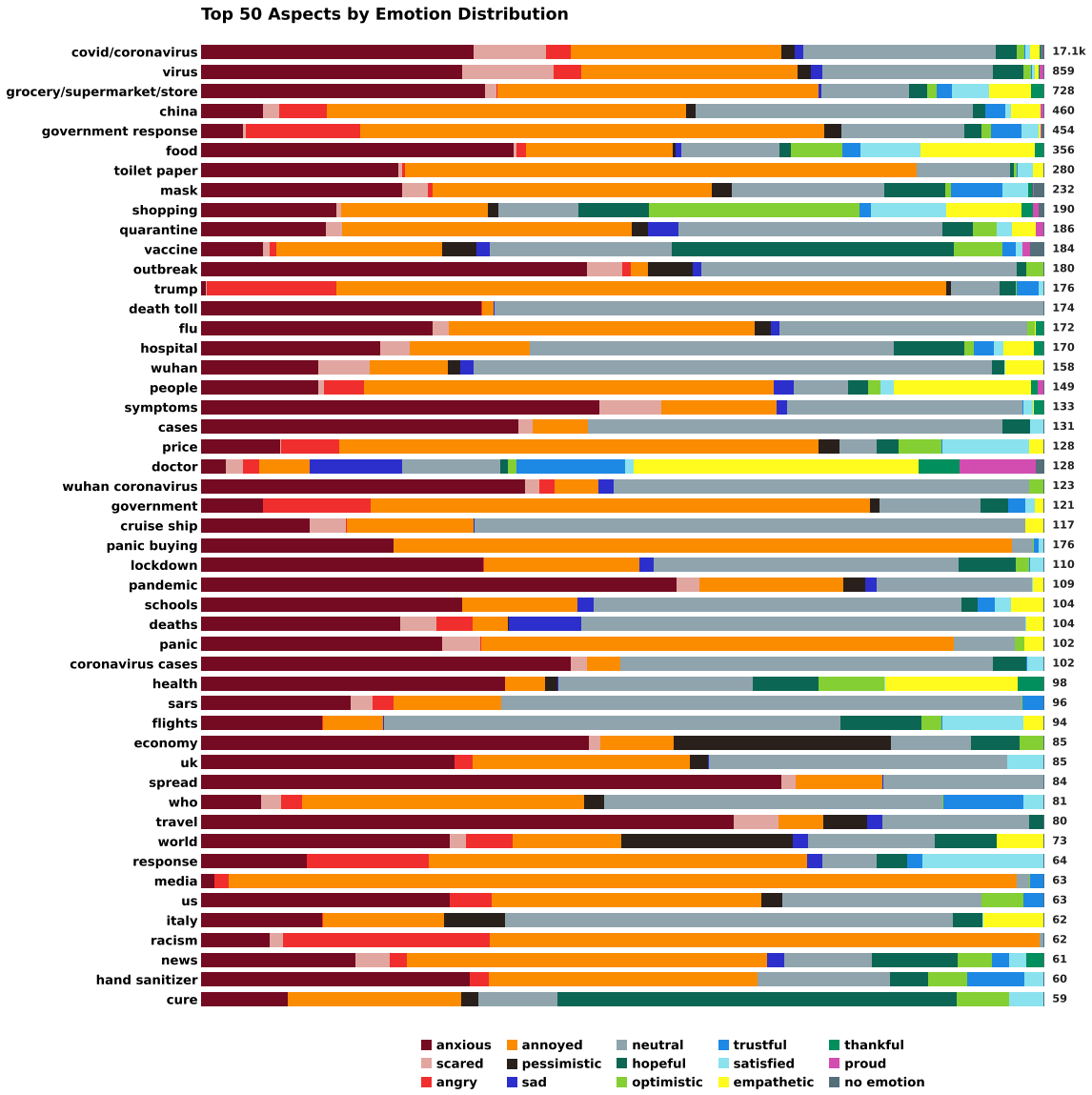


Figure 5: Emotion distribution across the 50 most frequent aspects. Each bar represents the normalized distribution of emotion categories for a given aspect.

Discourse-Structural Noise in LLM-Generated Explanations

Raghvi Baloni
University of Trier
Trier, Germany
baloni@uni-trier.de

Achim Rettinger
University of Trier
Trier, Germany
rettinger@uni-trier.de

Abstract

Large language models (LLMs) are increasingly utilized to generate explanations of complex phenomena, yet existing evaluation frameworks assess quality extensively through semantic similarity to human references, a measure blind to discourse organization. We conduct a computational study of 300 ELI5 question pairs, comparing Llama-3.1 8B, Gemma-2 9B, and Mistral-7B against human-written responses using PDTB-grounded connective analysis across five relation types. All three models exhibit elaborative dominance, i.e. Elaborative connectives account for 56-66% of discourse markers versus 38% in human writing, with Causal, Contrastive, and Conditional relations systematically suppressed. This 18-28 percentage point gap persists after connective sense disambiguation (11-20 pp), ruling out measurement artifact. Across three semantic similarity metrics, PDTB connective noise explains less than 8% of shared variance (r^2<0.08), confirming it captures evaluation signal invisible to existing approaches. We establish discourse-structural noise as a distinct, measurable evaluation dimension for LLM-generated explanations. [1]

1 Introduction

The usage of large language models as explanation generators has accelerated across domains ranging from medical question answering to educational tutoring (Fan et al., 2019). Evaluation of these explanations has primarily focused on semantic fidelity (does the model convey the right content?) as measured by reference-based metrics such as ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2020), or by LLM-as-judge frameworks such as G-Eval (Liu et al., 2023).

These frameworks do not measure *discourse structure*, the rhetorical organization of an explanation, explaining how claims relate to evidence, how contrasts are drawn, how conditions are introduced. Discourse theory has long established that the same propositional content can be organized in structurally very different ways, and that structural variation affects how readers process and evaluate text (Mann and Thompson, 1988; Prasad et al., 2008). If LLMs systematically produce explanations with a different discourse structure from human writers, this divergence is latent to current evaluation practice.

We introduce the concept of **discourse-structural noise** as the systematic deviation of LLM explanation structure from the discourse organization patterns present in human-authored explanations from the same communicative task. The human connective distribution constitutes the expected structural signal for explanation text in communicative context. Systematic deviation from it, however internally consistent across models, constitutes noise relative to that reference standard. This framing does not imply randomness, rather, the cross-model consistency of the deviation makes it diagnostically meaningful and systematically measurable. We operationalize this through PDTB-grounded connective analysis, which decomposes markers into five relation types (Temporal, Causal, Contrastive, Elaborative, Conditional) and measures the distributional distance between LLM and human profiles.

Our main findings are: **(i)** all three LLMs tested exhibit elaborative dominance, an 18-28 pp overuse of Elaborative connectives paired with systematic suppression of argumentative relation types, robust to connective sense disambiguation; **(ii)** this pattern is consistent across distinct model families, indicating it is a property of LLM explanation generation broadly rather than any individual model; **(iii)** PDTB connective noise explains less than 8% of shared variance with three semantic similarity metrics (r^2<0.08 across ROUGE-L, DistilBERT,

[1] https://github.com/RaghviBaloni/discourse-structural-noise

Proceedings of the 11th Workshop on Natural User-generated Text, pages 65–71
October 28, 2026 ©2026 Association for Computational Linguistics

and MiniLM), confirming it constitutes a distinct and currently unmeasured evaluation dimension.

2 Related Works

Discourse coherence evaluation The Penn Discourse Treebank (PDTB) (Prasad et al., 2008) provides the canonical framework for annotating discourse relations via connective expressions. Entity grid models (Barzilay and Lapata, 2008) capture local coherence through entity role transition across adjacent sentences. More recent neural approaches (Liu and Strube, 2025) extend coherence modeling with discourse relation supervision. These frameworks were developed and validated on human-authored newswire and academic text while their behavior on LLM-generated conversational explanation text remain uncharacterized.

LLM output quality evaluation BERTScore and ROUGE (Zhang et al., 2020; Lin, 2004) measure reference-based semantic similarity. G-Eval and GPT-Score (Liu et al., 2023; Fu et al., 2023) leverage LLM as judges of holistic quality dimensions including coherence, fluency, and relevance. (Wiegreffe and Marasović, 2021) survey explanation evaluation broadly and identify faithfulness and plausibility as dominant criteria, neither of which targets discourse-structural organization. Our work fills this gap by introducing a computationally grounded, theoretically motivated measure of structural divergence.

LLM-generated text characteristics Prior work has characterized LLM outputs in terms of lexical diversity, factual consistency, and stylistic properties (Shaib et al., 2024; Laban et al., 2021; Muñoz-Ortiz et al., 2024). Discourse-structural properties of LLM-generated explanations have not been systematically audited. Existing evaluation efforts either rely on costly human annotation (Wiegreffe and Marasović, 2021) or holistic LLM-as-judge frameworks, neither of which provides systematic, reproducible discourse-structural analysis at scale.

The work of Wang et al. (Wang et al., 2022) is directly complementary to ours, they study exemplification as a discourse role in long-form QA and show that model-generated answers under-use certain discourse functions relative to human writers, consistent with our finding that LLMs suppress argumentative relation types. We extend this observation from the document-level perspective to the connective-level distributional perspective, providing a finer-grained and automatically scalable measure of structural divergence.

3 Methodology

3.1 Dataset and Sampling

We sample 300 question-explanation pairs from the ELI5 dataset (Fan et al., 2019), a long-form question-answering benchmark sourced from r/explainlikeimfive subreddit. Each item consists of a community-posed question and one or more human-authored explanations scored by community upvotes. We retain the highest-scored answer per question and apply three quality filters: a minimum of 50 and maximum of 500 words, at least three sentences, and a minimum community score of 2. These thresholds ensure explanations are discourse-rich enough to support structural analysis while avoiding multi-topic responses that confound cross-explanation comparison.

3.2 Models

We evaluate three open-weight instruction-tuned language models spanning a range of parameter scales, Gemma-2 9B, Llama-3.1 8B, and Mistral-7B v0.3 (Team, 2024; Grattafiori et al., 2024; Jiang et al., 2023). For each of the 300 questions, we generate one explanation per model using vLLM (Kwon et al., 2023) with temperature 0.7, top-p 0.9, and a maximum of 512 output tokens. All models receive an identical system prompt instructing them to provide a clear, structured explanation, without specifying any discourse organization. This ensures that any structural patterns we observe are emergent properties of the models' generation behavior, not artifacts of explicit instruction.

3.3 Discourse-Structural Noise

We define **discourse-structural noise** as the systematic deviation of LLM-generated explanations from the discourse organization patterns present in human-written explanations for the same communicative task. We operationalize this along two dimensions:

PDTB Connective Analysis Based on the PDTB 2.0 sense hierarchy, we categorize discourse connectives into five relation types: Temporal, Causal, Contrastive, Elaborative, and Conditional. We compile lexicon of 120+ connective expressions across these five types, with multi-word expressions matched before their sub-strings to prevent spurious overlap. We categorize connectives

following the PDTB 2.0 coarse-grained sense hierarchy: TEMPORAL (eg. *then, after, when*), CONTRASTIVE (eg. *however, although, yet*), CONDITIONAL (e.g. *if, unless, provided that*). Multi-word expressions (e.g. *even though*) are matched before their constituent sub-strings (e.g. *though*) to prevent spurious double-counting. For each text, we extract all connective instances and compute a normalised distribution over the five types. We then measure PDTB connective noise as the symmetric KL divergence between each LLM's connective distribution and the human distribution across the 300 matched samples.

Let $\tilde{P}(t) = \big(P(t)+\varepsilon\big) \,/\, \sum_{t'}(P(t')+\varepsilon)$ denote the smoothed distribution with $\varepsilon = 10^{-8}$. We measure PDTB connective noise as the symmetric KL divergence between the smoothed LLM and human distributions:

$$D_{\text{sym}}(\tilde{P}_{\text{LLM}} \| \tilde{P}_{\text{H}}) = \frac{1}{2}\Big[\sum_t \tilde{P}_{\text{LLM}}(t) \log \frac{\tilde{P}_{\text{LLM}}(t)}{\tilde{P}_{\text{H}}(t)} + \sum_t \tilde{P}_{\text{H}}(t) \log \frac{\tilde{P}_{\text{H}}(t)}{\tilde{P}_{\text{LLM}}(t)}\Big] \quad (1)$$

where t indexes the five relation types.

Semantic similarity baseline To test whether discourse-structural noise captures evaluation signal orthogonal to existing semantic metrics, we compute a sentence-embedding cosine similarity score for each LLM response against its paired human explanation, using DistilBERT-base-uncased (Sanh et al., 2020) with mean pooling over non-padding tokens. We report Spearman rank correlation (ρ) between per-sample KL divergence scores and semantic similarity scores.

4 Results

4.1 LLMs Exhibit Consistent Elaborative Dominance

Figure 1 shows the mean normalised connective type distribution for human explanations and all three LLMs. Human explanations exhibit a relatively balanced discourse profile, with Causal connectives accounting for the largest share (25.0%), followed by Elaborative (37.9%), Temporal (14.7%), Contrastive (13.0%), and Conditional (9.1%).

LLM-generated explanations show a significantly different structural profile. Across all three models, Elaborative connectives dominate, accounting for 56.2% (Mistral) to 66.3% (Llama-3.1) of all connective instances, an increase of 18-28 percentage points over the human baseline. This shift is accompanied by systematic suppression of all argumentative relation types - Causal connectives drop from 25.0% in humans to 14.0-17.9% across models, Contrastive from 13.0% to 5.6-7.7% and Conditional from 9.1% to 2.2-3.2%.

Source	Total	TEMP	CAUS	CONT	ELAB	COND	KL↑
Human	11.2	14.7	25.0	13.0	37.9	9.1	—
Gemma-2 9B	18.9	14.7	15.6	5.6	60.8	3.2	2.39
Llama-3.1 8B	25.1	10.8	14.0	6.3	66.3	2.2	2.54
Mistral-7B	20.6	11.4	17.9	7.7	56.2	3.2	2.34

Table 1: Mean total connective count and normalised proportion (%) per relation type. KL = symmetric KL divergence from human profile; ↑ = higher means more structural noise.

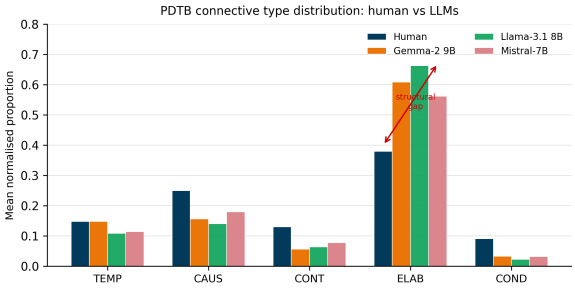


Figure 1: Mean normalised PDTB connective type distribution across 300 matched ELI5 pairs. All three LLMs show elaborative dominance (ELAB: 56-66%) versus a balanced human profile. Error bars show one standard deviation.

This pattern is consistent across all three model families tested. Gemma-2, Llama-3.1, and Mistral-7B, architecturally distinct models trained on different data mixtures, converge on the same discourse structure where elaborative addition replaces argumentative reasoning as the dominant connective mode. This consistency between models indicates that elaborative dominance is broadly a structural property of LLM explanation generation, rather than an idiosyncrasy of any particular model or training procedure.

Table 1 further shows that LLMs generate substantially more total connective instances than humans (19-25 vs 11 per text), yet this increase is concentrated entirely in elaborative connectives. The models are discourse-monotone instead of discourse-sparse, defaulting to a single relational register regardless of the communicative demands of the question. Bootstrap 95% confidence intervals confirm that the human-LLM gap on all four

argumentative relation types is non-overlapping across all models (Table 2 in Appendix), establishing the finding as statistically robust. Furthermore, surface features including response length explain only 1.9% of KL divergence variance (R^2=0.019; Appendix E), ruling out length and complexity as confounds.

4.2 Robustness to Connective Sense Disambiguation

A potential confound is connective polysemy, high frequency markers such as *since, as, while, and* carry multiple sense across PDTB relation types. To assess whether elaborative dominance is an artifact of mis-classification, we re-ran the connective analysis with dependency parse sense disambiguation using spaCy (`en_core_web_trf`), applying heuristics for the five most polysemous connectives.

The disambiguation reclassified 393, 568, 1112, and 819 connective instances for human, Gemma-2, Llama-3.1, and Mistral-7B responses respectively. After reclassification, Elaborative proportions shift to 30.9% for Human and 51.0%, 49.8%, 42.3% for Gemma-2, Llama-3.1, and Mistral-7B respectively. The human-LLM gap narrows from 18-28 to 11-20 percentage points but is not eliminated under any model. Notably, Llama-3.1 had the most connective instances reclassified yet maintained the highest elaborative proportion after disambiguation, indicating that its dominance is driven by unambiguously elaborative markers (*furthermore, additionally, also*) that are not subject to sense ambiguity. We conclude that polysemy partially explains but does not account for elaborative dominance.

4.3 Discourse-Structural Noise is Substantively Independent of Semantic Similarity

Figure 2 shows Spearman ρ between PDTB KL divergence and three semantic similarity metrics, ROUGE-L, DistilBERT cosine similarity, and MiniLM cosine similarity (Wang et al., 2020; Lin, 2004; Sanh et al., 2020), for each model. Full numerical results are reported in Appendix Table 3.

All correlations are negative and small in magnitude. The largest observed correlation (Llama-3.1 $\times$ ROUGE-L:ρ=-0.27, r^2=0.07) explains less than 8% of shared variance. Across all other model-metric combination, r^2<0.05, confirming that the two measures capture largely independent signal about explanation quality. Statistically significant

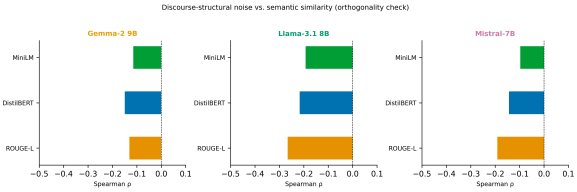


Figure 2: All bars extend leftward (negative ρ), confirming an inverse relationship; short bar lengths confirm that structural noise is largely independent of semantic similarity.

correlations (p<0.05) appear in the majority of combinations, as expected at N=300 for non-zero population correlations. We report effect sizes (r^2) as the relevant quantity.

The practical implication is direct. Mean semantic similarity scores are nearly identical across all three models (DistilBERT: 0.887-0.902, range <0.02), whereas KL divergence shows model-specific variation (2.34-2.54). An evaluation framework relying on semantic similarity alone would observe no meaningful difference between models and would systematically miss the structural deviation we document.

5 Conclusion

We introduce discourse-structural noise as a previously uncharacterized measurable failure mode of LLM explanation generation. Through a PDTB-grounded audit of 300 ELI5 question pairs across three open-weight model families, we identify consistent elaborative dominance. Dependency-parse sense disambiguation of the five most polysemous connectives reduces the elaborative gap to 11-20 pp but does not eliminate it. PDTB connective noise explains less than 8% of shared variance with three independent semantic similarity metrics, confirming it captures evaluation signal invisible to current metrics. Surface features including response length account for only 1.9% of KL divergence variance, ruling out length and lexical complexity as confounding explanations.

These findings have direct implications for LLM evaluation practice, discourse structure should be treated as an independent quality dimension alongside semantic fidelity and factual correctness. We release our PDTB connective evaluator, corpus of 300 human-LLM explanation pairs and code to support reproducibility.

Limitations

Connective lexicon and implicit relations Our rule-based lexicon may miss implicit discourse relations not signaled by surface connectives. Several high-frequency markers (*since*, *as*, *while*, *and*) carry multiple senses across PDTB relation types. We partially address this in §4.2 through dependency-parse sense disambiguation of the five most polysemous connectives (spaCy `en_core_web_trf`), which reduces but does not eliminate the elaborative dominance gap (11-20pp post-disambiguation vs. 18–28pp baseline). Implicit discourse relations, those not signaled by overt connectives, remain uncaptured. The observed suppression of Causal and Contrastive relations may partly reflect a shift toward implicit signaling rather than genuine argumentative impoverishment.

Prompt and length confounds Our system prompt requests a "clear, structured explanation," phrasing that may encourage additive enumeration. However, surface features including response length (ρ=−0.008, p=0.810), sentence count (p=0.947), and model identity together explain only 1.9% of KL divergence variance (R^2=0.019; Appendix E), providing empirical evidence against a length or prompt artifact account.

Scope This study covers three English-language instruction-tuned models and one domain (ELI5 long-form QA). Generalization to other model families, languages, and explanation domains remains to be established.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *Preprint*, arXiv:2302.04166.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Summac: Re-visiting nli-based models for inconsistency detection in summarization. *Preprint*, arXiv:2111.09525.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Wei Liu and Michael Strube. 2025. Discourse relation-enhanced neural coherence modeling. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4748–4762, Vienna, Austria. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

William Mann and Sandra Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text - Interdisciplinary Journal for the Study of Discourse*, 8:243–281.

Maximilian Maurer. 2026. elfen: A python package for efficient linguistic feature extraction for natural language datasets. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 61–74, Rabat, Marocco. Association for Computational Linguistics.

Alberto Muñoz-Ortiz, Carlos Gómez-Rodríguez, and David Vilares. 2024. Contrasting linguistic patterns in human and llm-generated news text. *Artificial Intelligence Review*, 57(10).

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Chantal Shaib, Yanai Elazar, Junyi Jessy Li, and Byron C Wallace. 2024. Detection and measurement of syntactic templates in generated text. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6416–6431, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team. 2024. Gemma.

Shufan Wang, Fangyuan Xu, Laure Thompson, Eunsol Choi, and Mohit Iyyer. 2022. Modeling exemplification in long-form question answering via retrieval. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2079–2092, Seattle, United States. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable natural language processing. *Preprint*, arXiv:2102.12060.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

A Appendix

A.1 Connective Density by Type

B Bootstrap Confidence Intervals

Source	TEMP	CAUS	CONT	ELAB	COND
Human	$0.147_{\pm 0.019}$	$0.250_{\pm 0.020}$	$0.130_{\pm 0.014}$	$0.379_{\pm 0.022}$	$0.091_{\pm 0.014}$
Gemma-2 9B	$0.147_{\pm 0.011}$	$0.156_{\pm 0.011}$	$0.056_{\pm 0.007}$	$0.608_{\pm 0.017}$	$0.032_{\pm 0.006}$
Llama-3.1 8B	$0.108_{\pm 0.011}$	$0.140_{\pm 0.011}$	$0.063_{\pm 0.009}$	$0.663_{\pm 0.019}$	$0.022_{\pm 0.005}$
Mistral-7B	$0.114_{\pm 0.011}$	$0.179_{\pm 0.013}$	$0.077_{\pm 0.009}$	$0.562_{\pm 0.024}$	$0.032_{\pm 0.007}$

Table 2: Mean normalised connective type proportions with 95% bootstrap confidence intervals (2,000 resamples). Subscript $\pm$ shows the CI half-width. Non-overlapping intervals between human and all LLMs on ELAB and CAUS confirm the elaborative dominance finding is statistically robust.

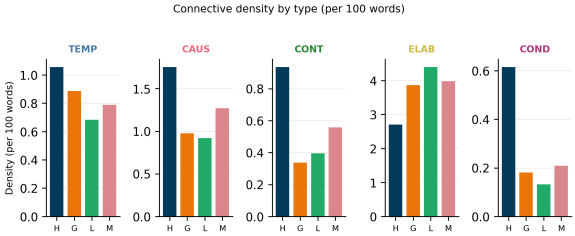


Figure 3: Mean connective density (instances per 100 words) by relation type for each source. Human explanations show higher density in Causal and Contrastive types relative to all LLMs. LLMs compensate with higher absolute Elaborative density, confirming the elaborative dominance finding at the rate level rather than only at the distributional level.

C Discourse-Structural Noise is Orthogonal to Semantic Similarity

Model	ROUGE-L		DistilBERT		MiniLM	
	ρ	p	ρ	p	ρ	p
Gemma-2 9B	-0.131	0.024^*	-0.149	0.010^*	-0.114	0.048^*
Llama-3.1 8B	-0.266	0.000^*	-0.217	0.000^*	-0.193	0.001^*
Mistral-7B	-0.191	0.001^*	-0.143	0.013^*	-0.097	0.092

Table 3: Spearman ρ between PDTB KL divergence and three semantic similarity metrics across 300 ELI5 pairs per model. Near-zero correlations across all three metrics confirm that discourse-structural noise captures evaluation signal orthogonal to existing semantic evaluation approaches. $^*p < 0.05$.

D Qualitative Examples

Table 4 presents representative examples from the highest-KL quartile of Llama-3.1 responses, illustrating how elaborative dominance manifests in practice.

E Surface Feature Analysis and Confound Control

To test whether elaborative dominance is an artifact of response length, lexical complexity, or model identity rather than a genuine structural property, we extracted eight surface linguistic features for all 1,200 texts (300 questions $\times$ 4 sources: human, Gemma-2, Llama-3.1, Mistral-7B) using ELFEN (Maurer, 2026), a multilingual linguistic feature extraction library, with a spaCy backbone.

Features extracted Token count, sentence count, tokens per sentence, average word length, type-token ratio (TTR), MTLD, Flesch reading ease, and token entropy.

Question	Human explanation (excerpt)	Llama-3.1 explanation (excerpt)
what bleach (clorox) does to organic things.	Bleach is most commonly sodium hypochlorite. The hypochlorite ion ClO^- is a very strong oxidizing agent, meaning that it steals the electrons off of other things. This is why it is such a good disinfectant, because ionizing (stealing electrons) from an organism's body is deadly. For things with a tiny body like bacteria, this is even more quickly fatal. *Connectives:* ***CAUSE*** *- because*	Bleach, specifically Clorox, is a strong chemical disinfectant that contains sodium hypochlorite (NaOCl) as its active ingredient. When bleach comes into contact with organic matter, it undergoes a series of chemical reactions that break down the organic material. Here's a step-by-step explanation of what bleach does to organic things: 1. **Penetration**: When bleach is applied to an organic surface, it penetrates the material, allowing the sodium hypochlorite molecules to come into contact with the organic matter. 2. **Oxidation**: The sodium hypochlorite molecules in bleach are strong oxid... *Connectives:* ***CONT*** *- however,* ***ELAB*** *- specifically, also,* ***TEMP*** *- then, when*
Who was allowed to vote in 19th century Britain?	In the 1800s suffrage was expanded about every 2 or 3 decades. At the start of the 19th century, probably 5% of the British population could vote. This right was reserved to wealthy landowners. The Reform Act of 1832 changed this. Voting rights were now spread to bankers, merchants, etc. The Reform Act of 1867 was the next time voting rights were expanded. If you were a male, owned a home, or rented a home of at least a certain value you were permitted to vote. The Reform Act of 1884 extended suffrage to those owning or renting land in the countryside, outside of town limits. Around the ti... *Connectives:* ***COND*** *- if*	To understand who was allowed to vote in 19th century Britain, we need to break down the historical context and the various reforms that took place during this period. **Early 19th century: Limited Voting Rights (1800s)** In the early 19th century, voting rights in Britain were restricted to a small group of people known as the "Electoral Classes." These included: 1. **Male property owners**: Free men who owned property, such as land or a house, worth a certain amount, were eligible to vote. This meant that only those with a certain level of wealth and social status could participate in th... *Connectives:* ***ELAB*** *- in addition, also*

Table 4: Representative high-KL divergence examples from Llama-3.1 8B. Human explanations use causal and contrastive connectives to build logical arguments; LLM explanations substitute elaborative connectives (*furthermore*, *additionally*, *also*), adding information without establishing argumentative relations.

Regression We fit an OLS regression with KL divergence as the outcome and the eight surface features plus model identity dummy variables as predictors. Results are reported in Table 5.

Predictor	ρ with KL	p
Token count	-0.008	0.810
Sentence count	$+0.002$	0.947
Tokens per sentence	(collinear)	
Type-token ratio	-0.084	0.012^*
Flesch reading ease	$+0.098$	0.004^*
Token entropy	$+0.025$	0.464
Model: Gemma-2	$+0.025$	0.454
Model: Llama-3.1	$+0.017$	0.611
Model: Mistral-7B	-0.043	0.204
$R^2 = 0.019,\ R^2_{\text{adj}} = 0.010,\ N = 889$		

Table 5: OLS regression of PDTB KL divergence on surface features and model identity dummies. Spearman ρ between each predictor and KL divergence; R^2 for the full model. Response length and sentence count are non-significant ($p > 0.80$); the full model explains only 1.9% of KL variance. $^*p < 0.05$.

Interpretation The full model explains only R^2=0.019 of KL divergence variance. Response length is not a significant predictor (ρ=-0.008, p=0.810), nor is sentence count (p=0.947). The two significant individual predictors, TTR (ρ=-0.084) and Flesch reading ease (ρ=$+0.098$), have trivially small effect sizes. Model identity dummy variables are non-significant after controlling for surface features (all p>0.20), consistent with the observation that all three models show high structural noise relative to the human baseline. These results confirm that elaborative dominance is not reducible to surface-level response characteristics and reflects genuine discourse-structural divergence.

Do LLMs Give Consistent Opinions? Evaluating Response Reliability Under Varying Likert-Scale Formulations in Survey-Style MCQA

Martin Kandlinger[*♣] **Bolei Ma**[*♣,♡,◇]
Anna-Carolina Haensch[♣,♡,◇] **Matthias Aßenmacher**[♣,♡]

[♣]LMU Munich [♡]Munich Center for Machine Learning
[◇]University of Maryland, College Park

Abstract

Survey methodology requires that equivalent question formulations elicit stable responses from the same respondent, regardless of how the Likert-type answer scale is presented. As LLMs are increasingly deployed as synthetic survey respondents, a critical but underexplored question arises: do they satisfy this basic *parallel-forms reliability* standard when answer scales vary in wording, cardinality, polarity, or option order? We extend the OpinionQA into two stress-test benchmarks, 1,235 subjective questions each paired with five reworded scale variants and six positional permutations, evaluated across four prompt-label formats. Using a four-metric consistency framework, we benchmark eight open-weight 7B-9B models from four families in both base and instruction-tuned configurations. Three findings emerge. First, base models fail basic reliability requirements: they exhibit severe mechanical instability driven by primacy bias, label sensitivity, and formatting failures, including full polarity inversions for the same question. Second, instruction tuning acts as a reliability filter, reducing response variance by approximately 50% and enabling models to treat semantically equivalent scales as equivalent. Third, for instruction-tuned models under randomized option ordering, removing explicit labels paradoxically *increases* consistency by eliminating symbol-binding conflicts. These findings establish that response reliability is an engineered property of LLMs requiring both alignment training and deliberate prompt design, with concrete implications for survey-like MCQA evaluations. [1]

1 Introduction

The reliability of survey measurement instruments is a cornerstone of empirical social science. Classical psychometric theory formalizes this through the concept of *parallel-forms reliability*: equivalent versions of a survey question, differing only in surface-level scale properties, should produce comparable responses from the same respondent (Lord et al., 1968; Cronbach, 1951). As LLMs are increasingly deployed as synthetic survey respondents (Aher et al., 2023; Santurkar et al., 2023), this standard becomes directly testable. Because subjective survey items lack a ground-truth "correct" answer, standard measures of *validity* (or accuracy) are undefined. Therefore, this paper strictly evaluates *reliability* (self-consistency). A reliable LLM respondent should express the same underlying opinion regardless of whether the scale uses four or five points, whether agreement is worded as "Strongly" or "Fully", or whether the positive end of the scale appears first or last. Figure 1 shows an example of an LLM responding to the same question under different scale configurations.

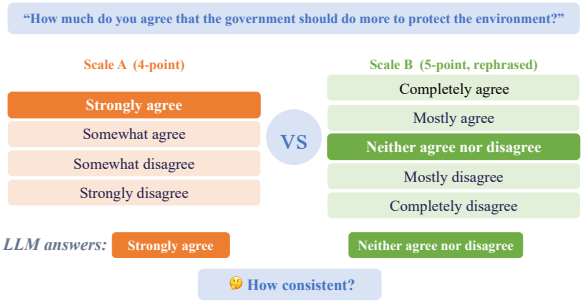


Figure 1: Example of LLMs in responding to a question with different scales. Are they consistent?

Evidence on LLM format sensitivity suggests this requirement is far from guaranteed. Models exhibit systematic positional biases (Zheng et al., 2024; Wang et al., 2024a), fail to robustly bind answer content to option labels (Xue et al., 2024), and produce volatile outputs under minor prompt perturbations (Ngweta et al., 2025; Lunardi et al., 2025). Several studies document that LLM survey responses shift substantially with scale redesign

[*]Equal contributions. Contact: bolei.ma@lmu.de
[1]Data and code are available at: https://github.com/boleima/SurveyScaleQA.

Proceedings of the 11th Workshop on Natural User-generated Text, pages 72–88
October 28, 2026 ©2026 Association for Computational Linguistics

(Rupprecht et al., 2025; Khan et al., 2025; Lee et al., 2025). However, these studies typically examine a single model family, do not manipulate scale design as systematically as survey researchers would, and rarely isolate the role of alignment training as a driver of reliability.

This paper addresses these gaps through a controlled, multi-factor evaluation grounded in survey methodology. We vary three dimensions of Likert-scale design: (1) semantic wording and cardinality, (2) option order, and (3) prompt-label format. We evaluate all combinations across eight models from four families, comparing Base and Instruction-Tuned variants. Crucially, to strictly isolate the effects of prompt format and scale design from stochastic generation noise, we enforce greedy decoding across all experiments; our measurement of "consistency" therefore captures deterministic structural sensitivity rather than sampling variance.

Contributions. This paper makes the following contributions:

- We release an extended OpinionQA benchmark with systematic Likert-scale variations (6,175 Reworded and 7,410 Shuffled question-scale pairs, each across four prompt formats), designed as a reusable consistency evaluation resource.
- We introduce a four-metric consistency framework that disentangles average drift (APD, MSD), worst-case divergence (MPD), and cross-format structural stability (Pearson r), grounded in the psychometric concept of parallel-forms reliability.
- We provide the first large-scale comparison of Base vs. Instruction-Tuned models across Likert scale design dimensions, establishing that for models in the 7B-9B parameter class, alignment is the primary reliability determinant and reduces response variance by $\approx$50%.
- We show a counterintuitive pattern: when option order is randomized, removing labels actually makes instruction-tuned models *more* consistent, not less, because it avoids a conflict between a model's habitual label choice and the option it actually means to pick, an interaction between prompt format and option order that prior work has largely overlooked.

2 Background and Related Work

Survey reliability and Likert-scale design. Classical Test Theory decomposes an observed survey response into a true latent score and a measurement error component (Lord et al., 1968). A key source of measurement error in Likert-scale surveys is scale design itself: the number of response options, the directionality (unipolar vs. bipolar), and verbal labels attached to scale points have all been shown to shift human responses independently of the underlying attitude (Menold, 2021; Höhne et al., 2022). Human respondents also exhibit well-documented response styles, including acquiescence bias (Couch and Keniston, 1960), satisficing (Krosnick, 1991), and order effects (Krosnick and Alwin, 1987). Parallel-forms reliability addresses this challenge by requiring that two equivalent instrument versions, differing only in surface properties, yield correlated results (Cronbach, 1951). This is precisely the standard we apply to LLMs: a model queried with semantically equivalent scales should express the consistent underlying opinion.

LLM format sensitivity and symbol binding. Accumulating evidence shows that LLM responses in multiple-choice settings are highly sensitive to presentation format. Models exhibit primacy and recency biases, with GPT-3.5's accuracy on MMLU dropping by over 6 percentage points when the correct answer is consistently placed last (Zheng et al., 2024). First-token probabilities and text-output responses frequently disagree for the same prompt, with mismatch rates exceeding 60% for instruction-tuned models (Wang et al., 2024b), and minor formatting changes can cause accuracy fluctuations exceeding 76% (Ngweta et al., 2025). Xue et al. (2024) trace these failures to inadequate *symbol binding*, namely the inability to reliably associate option content with its assigned label, a failure mode that persists through fine-tuning. Molfese et al. (2025) further show that evaluation procedures can confound reasoning failures with formatting failures, underscoring the need to evaluate response consistency separately from accuracy.

LLMs as synthetic survey respondents. When LLMs act as virtual survey respondents, format sensitivity becomes a validity threat. Rupprecht et al. (2025) systematically perturb World Values Survey items and find consistent human-like biases, including strong recency effects, in LLM responses. Khan et al. (2025) show that apparent cultural alignment signals are largely methodological artifacts of prompt variation rather than stable model traits. Tjuatja et al. (2024) find that LLMs generally fail

to replicate human response biases in controlled survey settings, and Lee et al. (2025) demonstrate that even high-performing models are not consistent evaluators across rating scales. Santurkar et al. (2023) found that base models tend toward liberal-leaning opinions and are difficult to steer toward specific demographic views, a finding whose implications for opinion stability this paper directly addresses.

3 Methodology

3.1 Dataset Construction: How We Vary the Scales

Source Data. We build on the OpinionQA corpus (Santurkar et al., 2023), derived from the Pew American Trends Panel (Pew Research Center, 2024). After manually filtering questions requiring specific respondent personas or responses that cannot be mapped to Likert-type scales, we retain $N = 1{,}235$ highly subjective questions. Each question is classified into one of 15 semantic *content dimensions* (e.g., Agreement, Importance, Frequency, Better/Worse), which determine which family of Likert scales is appropriate for generating variations. The full distribution is in Appendix A.

Reworded Dataset: same question, different scale. For every question, we write five answer scales that all ask the same thing but differ along three dimensions that survey designers routinely choose between: how many response points the scale has (**cardinality**: 4, 5, or 6 points), which words label those points (**phrasing**, e.g., "Strongly" vs. "Fully" vs. "Completely"), and whether the scale runs from negative to positive or from zero to high (**polarity**: bipolar vs. unipolar). Table 1 shows this concretely for one question: V1 and V2 are both 4-point bipolar scales that only swap the wording ("Strongly/Somewhat" vs. "Fully/Partially"), while V3 keeps similar wording but moves to five points and adds a neutral middle option. Repeating this across all 1,235 questions gives a Reworded Dataset of $6{,}175$ question-scale pairs ($1{,}235 \times 5$), where the question itself never changes and only the scale wrapped around it does.

Shuffled Dataset: same scale, different order. To isolate the effect of option order on its own, we take one representative scale per question and generate five random permutations of its option order, using fixed seeds (2, 7, 10, 67, 101), and then add the original ordering back as a sixth version. The bottom row of Table 1 (S1) shows one such shuffle of V1: the wording and number of options are identical to V1, but "Strongly agree" now sits in the third position instead of the first. This gives a Shuffled Dataset of $7{,}410$ pairs ($1{,}235 \times 6$), where wording and scale length are held fixed and only the position of each option varies.

Response Normalization. Because we need to compare answers across scales with different numbers of points, we map every valid response onto the same $[-1, 1]$ interval, where $+1.0$ marks the most positive option (e.g., "Strongly Agree") and -1.0 the most negative. Options are spaced evenly along this interval: a 4-point scale maps to $\{-1.0, -0.33, 0.33, 1.0\}$, a 5-point scale to $\{-1.0, -0.5, 0.0, 0.5, 1.0\}$, and so on, as shown in the rightmost column of Table 1. This lets us measure how far apart two answers are even when they come from scales of different lengths.

Illustrative Example. Table 1 traces the full pipeline for one OpinionQA question. The question stem is held constant throughout. Three reworded variants illustrate how cardinality, phrasing, and polarity are varied: V1 and V2 are both 4-point bipolar scales with synonymous labels, while V3 extends to five points, adding a neutral midpoint. All three map semantically equivalent top options to $+1.0$, so a consistent model should select $+1.0$ across all variants regardless of the specific wording. The final row shows a shuffled permutation of V1: here, the option previously in position 1 ("Strongly agree") now occupies position 3. A model relying on positional cues would select option 1 in both cases, yielding $+1.0$ in V1 but -0.33 in S1, and an APD of 1.33 for this single question.

3.2 Model Configurations

We evaluate four open-weight model families, including Llama 3.1 (Grattafiori et al., 2024), Mistral (Jiang et al., 2023), Qwen 2.5 (Qwen et al., 2025), Gemma 2 (Team et al., 2024), each in both a pretrained Base and an Instruction-Tuned (IT) variant. All models range from 7B to 9B parameters, balancing computational feasibility with broad representativeness across pre-training recipes and tokenization strategies. Comparing Base and IT variants within each family isolates the effect of alignment training while holding architecture and scale constant.

All inference uses greedy decoding (`do_sample=False`) to eliminate sampling

Variant	Options (in presentation order)	Normalized values
Question: "How much do you agree that the government should do more to protect the environment?"		
V1 (4-pt, bipolar)	Strongly agree / Somewhat agree / Somewhat disagree / Strongly disagree	$+1.0$ / $+0.33$ / -0.33 / -1.0
V2 (4-pt, bipolar)	Fully agree / Partially agree / Partially disagree / Fully disagree	$+1.0$ / $+0.33$ / -0.33 / -1.0
V3 (5-pt, bipolar)	Completely agree / Mostly agree / Neither / Mostly disagree / Completely disagree	$+1.0$ / $+0.5$ / 0.0 / -0.5 / -1.0
S1 (V1, shuffled)	Somewhat disagree / Strongly disagree / **Strongly agree** / Somewhat agree	-0.33 / -1.0 / $+1.0$ / $+0.33$

Table 1: Variation Engine example. V1 and V2 use different labels for the same 4-point bipolar structure; V3 adds a neutral midpoint. All three map the top option to $+1.0$. S1 shuffles V1: a primacy-biased model selecting option 1 receives -0.33 instead of $+1.0$, yielding APD $= 1.33$ for this question.

noise, ensuring that any observed response variation is attributable to prompt design rather than stochastic generation. Details are shown in Appendix A.

3.3 Prompting Strategies

We evaluate four zero-shot prompt formats as a cross-cutting experimental factor, designed to test robustness to structural label choices that vary across real survey instruments:

- **P1 – Numeric:** Options labeled "1), 2), 3), . . . "
- **P2 – Alphabetic:** Options labeled "A), B), C), . . . "
- **P3 – Unlabeled:** Options listed as plain text lines, with no labels.
- **P4 – Extended:** Numeric labels augmented with two non-substantive options: "Don't Know" and "Refused", mimicking real survey instruments that include refusal codes.

IT models receive all instructions via a single `user` turn in the model's chat template, with no separate system prompt (to standardize across model families). Base models receive a raw text-completion prompt ending with a forcing suffix (e.g., "The best answer is option number:"). Full prompt templates are given in Appendix B.

3.4 Consistency Metrics

Rather than measuring accuracy against a ground truth, which is undefined for subjective opinions, we quantify *self-consistency* at the question level. This mirrors the psychometric concept of parallel-forms reliability: a model expressing a stable underlying opinion should produce responses that cluster tightly regardless of which scale formulation is used. For a set of valid normalized responses $V = \{v_1, \ldots, v_n\}$ ($n = 5$ for Reworded, $n = 6$ for Shuffled), we compute four metrics:

Average Pairwise Distance (APD). The mean absolute difference across all unique response pairs. APD $= 0$ indicates perfect consistency; higher values indicate semantic drift across scale variants.

$$\text{APD} = \frac{2}{n(n-1)} \sum_{i<j} |v_i - v_j| \tag{1}$$

Maximum Pairwise Distance (MPD). The worst-case divergence, $\max_{i<j} |v_i - v_j|$. An MPD value of 2.0 represents a complete polarity inversion, i.e., a shift from the most positive response extreme to the most negative for the same question.

Mean Standard Deviation (MSD). The mean of per-question standard deviations across all questions in the dataset. MSD captures average response dispersion and is more sensitive to general instability than to individual extreme cases.

Pearson Correlation (r). Computed between model response vectors across prompt-format pairs (e.g., P1 vs. P2). A high r indicates that the model preserves the relative semantic ranking of questions even when the absolute response format changes, reflecting structural stability of the model's internal opinion ordering.

Responses failing strict output parsing and non-substantive P4 selections ("Don't Know", "Refused") are excluded. A question is included only if all n variations yielded valid substantive responses, to avoid artificially deflating variance through missing data.

4 Results

4.1 Global Response Patterns: Mechanical Artifacts vs. Semantic Stability

Before computing consistency metrics, we examine the raw response distributions to characterize the systematic biases present in each model class. Figure 2 shows the percentage of responses in each option category for Base models on the Reworded Dataset.

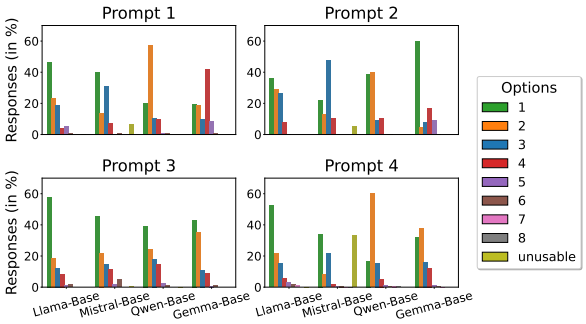


Figure 2: Response category distributions for Base models on the Reworded Dataset. The four subplots within each model panel correspond to the four prompt formats (P1–P4). Valid option indices range from 1 to 6 (up to 8 in P4). Llama-3.1-8B shows consistent primacy bias; Gemma-2-9B shows radical label sensitivity (strong preference for Option 4 in P1, shifts to Option 1 in P2); Mistral-7B-v0.3 accumulates high rates of unusable outputs in P4; Qwen2.5-7B shows moderate and relatively stable preference patterns.

The figure reveals four qualitatively distinct failure modes in Base models, each reflecting a different source of measurement error from a survey methodology perspective:

Primacy bias (Llama-3.1-8B). This model defaults to Option 1 in 35–60% of all responses, depending on the prompt format, regardless of question content. This mirrors the primacy effects documented in human survey research (Krosnick and Alwin, 1987), but in an extreme, mechanically consistent form.

Label sensitivity (Gemma-2-9B). This model strongly favors Option 4 ($>$40%) under numeric labels (P1), but shifts dramatically to Option 1 ($>$60%) under alphabetic labels (P2). The preference is not driven by semantic content but by how option labels interact with the model's learned probability distribution over tokens.

Formatting failures (Mistral-7B-v0.3). This model produces unusable responses at elevated rates in P1 and P2 ($\approx$8%) and catastrophically in P4 ($\approx$35%), where the unusable category rivals the most selected valid option. This reflects a mismatch between the model's text-completion training and the structured output format required by survey questions.

Aggregate distributional instability. A reliable survey respondent should express a similar aggregate opinion profile regardless of whether options are presented in their standard order or shuffled. In contrast, Gemma-2-9B Base shows a near-complete inversion of its aggregate opinion distribution between the Reworded (fixed order) and Shuffled conditions: under the Reworded Prompt 1, density is concentrated at the negative extreme (-1.0); under the Shuffled condition, density spikes at the positive extreme ($+1.0$). This polarity flip, driven purely by option reordering, constitutes a fundamental reliability failure under the parallel-forms standard.

In contrast, IT models show near-perfect format adherence (unusable rates $<$0.1%), maintain stable response profiles across semantic rewordings, and produce aggregate opinion distributions that overlap substantially between Reworded and Shuffled conditions, demonstrating that alignment successfully grounds outputs in semantic content rather than presentation artifacts.

4.2 Intra-Prompt Consistency: Volatility Analysis

We now quantify how much a model's response to a given question changes across scale variations within a fixed prompt format.

Average and worst-case divergence. Figure 3 shows APD distributions for all models on the Shuffled Dataset under P1. Base model distributions are markedly diffuse: Mistral-7B-v0.3 spreads density nearly uniformly from APD 0.0 to 1.25 (mean $\approx$0.57), and Gemma-2-9B has a distribution shifted further right, with a median of $\approx$0.63 and a distinct density spike near 0.8. For comparison, an APD of 0.67 corresponds to a model that, on average, moves one full step on a 4-point scale between scale variants. Llama-3.1-8B and Qwen2.5-7B show somewhat better clustering in the 0.2–0.5 range, but retain substantial high-distance tails.

The worst-case metric (MPD) adds further evidence of Base model fragility. Across all Base models on the Reworded Dataset, a non-trivial fraction of questions receive MPD $= 2.0$, representing a complete polarity inversion where the model gives diametrically opposite responses (e.g., "Strongly Agree" and "Strongly Disagree") to the same question under different scale formulations. For Gemma-2-9B, a bimodal MPD distribution is observed, with density spikes at both 0.0 (perfect stability for some questions) and 2.0 (full inversion for others), indicating that the model's stability is entirely content-dependent with no general consistency principle.

Comparison to a random-response baseline. To determine whether Base models are merely out-

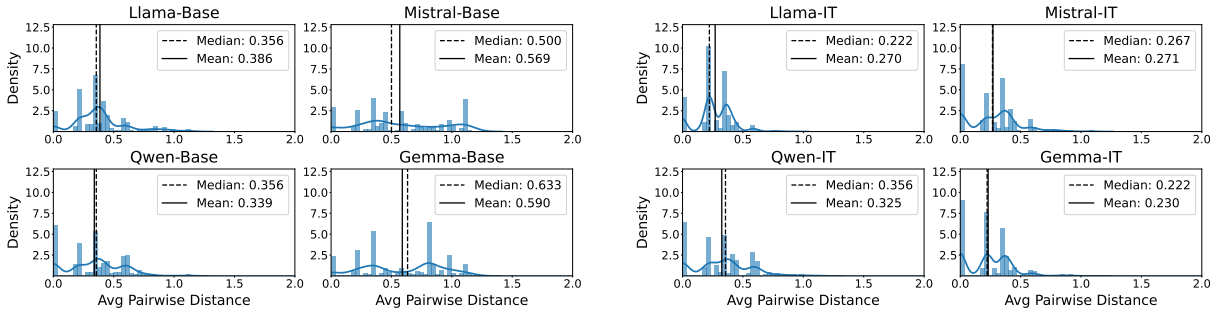


Figure 3: APD distributions on the Shuffled Dataset under P1. **Left**: Base models show highly diffuse distributions, reflecting continuous semantic drift across option permutations. **Right**: Instruction tuning compresses distributions sharply toward 0. The improvement is most dramatic for Gemma-2-9B, which transforms from the most volatile Base model to the most stable IT model.

putting noise, we simulated a uniform-random responder that selects valid scale options with equal probability. On the Reworded Dataset (5 variants), this null baseline yields a mean APD of 0.825, a mean MPD of 1.632, and a MSD of 0.711. On the Shuffled Dataset (6 variants), the mean APD is 0.833, mean MPD is 1.734, and MSD is 0.724. Comparing these null values to the observed Base model metrics (e.g., Mistral's Shuffled APD of ≈0.57 in Figure 3, or the Reworded MSDs of 0.22-0.45 shown in Table 2) confirms that Base models are not merely outputting uniform noise. They retain a meaningful semantic signal on average, albeit a weaker and more volatile one than that of IT models.

Model	P1	P2	P3	P4
Base Models				
Llama-3.1-8B	0.357	0.315	0.411	0.340
Mistral-7B-v0.3	0.302	0.254	0.453	0.294
Qwen2.5-7B	0.254	0.304	0.415	0.224
Gemma-2-9B	0.344	0.325	0.421	0.229
IT Models				
Llama-3.1-8B-IT	0.207	0.229	0.248	0.203
Mistral-7B-IT	0.221	0.227	0.287	0.206
Qwen2.5-7B-IT	0.232	0.249	0.244	0.241
Gemma-2-9B-IT	0.174	0.172	0.178	0.164

Table 2: Mean Standard Deviation (MSD) on the Reworded Dataset. P1 (Numeric), P2 (Alphabetic), P3 (Unlabeled), P4 (Extended). IT models achieve approximately half the MSD of Base models across all conditions.

The reliability filter effect of instruction tuning. Figure 3 (right) and Table 2 together quantify the stabilization produced by instruction tuning. IT model APD distributions concentrate sharply near 0 across all four families. Table 2 shows that across all models and prompt formats, IT models achieve MSD values roughly half those of their Base counterparts (e.g., Llama-3.1-8B drops from 0.357 to 0.207 under P1; Gemma-2-9B drops from 0.344 to 0.174). Gemma-2-9B-IT achieves the lowest MSD of all models (≈0.17 on average), a particularly striking result given that Gemma-2-9B (Base) is among the most volatile, confirming that stability is a consequence of alignment rather than an inherent architectural property.

Prompt format and label effects. An important secondary finding from Table 2 concerns the Unlabeled format (P3). For Base models, removing option labels causes MSD to spike substantially (Mistral-7B-v0.3: 0.453; all others: ≥0.41), confirming that labels function as structural anchors that support mapping semantic intent to a discrete output token. Without these anchors, Base models lose the ability to produce consistent selections. IT models are significantly less affected (P3 MSD: 0.178–0.287), reflecting their stronger semantic grounding.

4.3 Removing Labels Can Help When Options Are Shuffled

Comparing the Reworded and Shuffled datasets turns up a pattern that runs against intuition: whether unlabeled options help or hurt consistency depends entirely on whether the option order is fixed or scrambled.

On the Reworded Dataset, where option order is fixed, P3 (Unlabeled) produces the highest MSD of the four formats for both model classes, as discussed above, so dropping labels there clearly hurts. But on the Shuffled Dataset, where option order is randomized, P3 gives the *lowest* MSD of the four formats for every instruction-tuned model

family. As shown in Table 6 (Appendix C.5), Gemma-2-9B-IT achieves MSD = 0.121 under P3 on the Shuffled Dataset, lower than its values for P1 (0.232), P2 (0.187), and P4 (0.220). Similar patterns hold for Llama-3.1-8B-IT (P3: 0.146 vs. P1: 0.277) and Qwen2.5-7B-IT (P3: 0.206 vs. P1: 0.310).

Here is how we read this. Under labeled prompts (P1, P2) with shuffled options, instruction-tuned models are pulled in two directions at once: a learned habit of favoring certain labels (say, a tendency to answer "A" or "1" regardless of content) and a preference for whichever option actually matches their opinion. When shuffling happens to put the model's preferred content behind an unfavored label, these two pulls disagree, and the resulting answers become less stable. Take the labels away, and that conflict simply disappears: the model has nothing left to lean on but the content itself, and the content is the same no matter how the options are ordered, so consistency goes up.

This lines up with the Multiple Choice Symbol Binding account of Xue et al. (2024): the failure sits at the connection between a label and its content, so removing the label sidesteps the failure for models that understand the content well enough to rely on it. Base models do not show this effect, because they never developed strong enough content understanding to make use of the missing labels, so P3 stays their worst format either way.

4.4 Inter-Prompt Structural Robustness

We now assess whether models preserve a consistent logical stance when the structural format of the prompt itself changes, independent of option order and wording variations.

Categorical consistency. Figure 4 shows the distribution of categorical agreement levels across all four prompt formats on the Reworded Dataset. Perfect agreement ("4 Same") means the model selects the same response category for all four prompt formats for a given question.

Base models exhibit clear logical fragmentation. Gemma-2-9B is the most fragile: "2 Same" is its dominant category (≈45%), and it records the highest rate of total disagreement ("4 Different": ≈5%). Mistral-7B-v0.3 achieves perfect categorical agreement in fewer than 20% of questions. Llama-3.1-8B and Qwen2.5-7B perform somewhat better, with "4 Same" around 32%, but "3 Same" or fewer is still the majority outcome.

Instruction tuning induces a decisive shift toward structural stability. "4 Same" becomes the dominant outcome for all four IT families, ranging from 43% (Mistral-7B-IT) to 62% (Llama-3.1-8B-IT). The "4 Different" category drops to <1% for all IT models. The residual "3 Same" category (≈27–34%) primarily reflects structurally valid behavior: when a model appropriately selects "Don't Know" or "Refused" in P4, this prevents a four-way match even though the model's substantive response in P1–P3 is perfectly consistent.

Correlational stability. Table 3 presents Pearson's r between response vectors across all prompt-format pairs on the Shuffled Dataset, measuring whether the model's relative opinion ranking across questions is preserved under format changes.

Model	1–2	1–3	1–4	2–3	2–4	3–4
Base Models						
Llama-3.1-8B	0.545	0.354	0.594	0.441	0.511	0.413
Mistral-7B-v0.3	0.309	0.183	0.496	0.191	0.269	0.244
Qwen2.5-7B	0.608	0.428	0.849	0.431	0.572	0.400
Gemma-2-9B	0.468	0.155	0.499	0.312	0.557	0.378
IT Models						
Llama-3.1-8B-IT	0.797	0.636	0.796	0.686	0.778	0.644
Mistral-7B-IT	0.730	0.709	0.901	0.627	0.728	0.718
Qwen2.5-7B-IT	0.886	0.783	0.921	0.777	0.882	0.780
Gemma-2-9B-IT	0.849	0.733	0.856	0.784	0.824	0.717

Table 3: Pearson's r between prompt-format pairs on the Shuffled Dataset. IT models maintain high logical ranking stability ($r \approx 0.70$–0.92). Base models suffer severe degradation, especially in pairs involving P3. The P1–P4 pair is consistently the most stable, reflecting structural similarity.

IT models maintain high structural stability ($r \approx 0.70$–0.92) regardless of format pair. Base models' response vectors are substantially scrambled when format changes ($r \approx 0.15$–0.85). Two consistent structural patterns emerge. First, P1–P4 yields the highest correlations across all models: for IT models, r ranges from 0.796 to 0.921, and even for Base models Qwen2.5-7B achieves 0.849. This confirms that models recognize the structural similarity between numeric and extended-numeric formats and maintain opinions when refusal options are added. Second, prompt pairs involving P3 consistently yield the lowest correlations, often 0.1–0.2 points below labeled pairs, confirming that unlabeled prompts require a qualitatively different processing pathway.

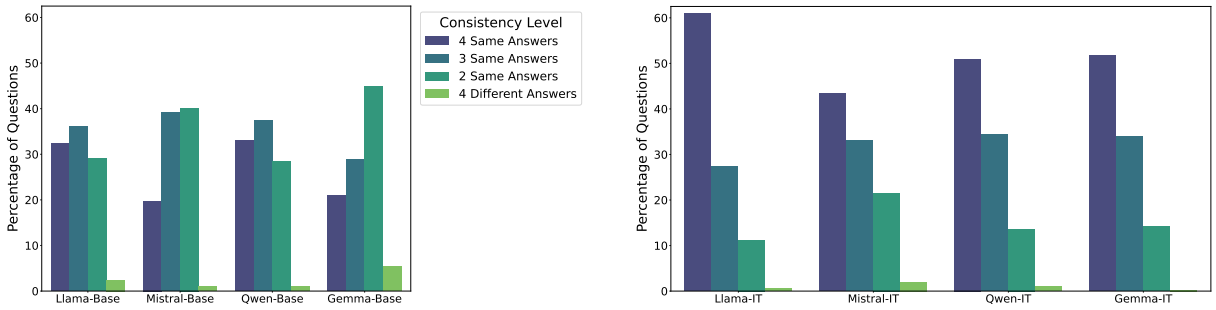


Figure 4: Categorical consistency across four prompt formats on the Reworded Dataset. **Left**: Base models exhibit logical fragmentation, with perfect agreement rarely dominant. **Right**: Instruction tuning shifts the dominant outcome to perfect agreement across all four families. The "3 Same" residual in IT models largely reflects structurally valid selections of "Don't Know" or "Refused" in P4.

5 Discussion

5.1 Psychometric Interpretation: Reliability as an Engineered Property

From the perspective of Classical Test Theory, our results can be summarized concisely: Base models fail to meet the parallel-forms reliability standard, while IT models approach it but do not fully achieve it. The measurement error introduced by scale design is substantial for Base models (MSD 0.22–0.55), meaning that a large fraction of the observed variation in Base model survey responses reflects properties of the measurement instrument rather than the model's underlying stance. IT models reduce this error substantially, but residual scale sensitivity (MSD 0.16–0.30) means that even aligned models do not yet meet the reliability standards typically required for survey instruments in social science research.

Importantly, this distinction is not primarily a matter of model architecture or parameter count. Within each of the four families, the IT variant dramatically outperforms the Base variant on every consistency metric. The effect of alignment training (SFT + RLHF) is a fundamental behavioral shift: Base models optimize for next-token prediction and are therefore sensitive to surface-level statistical regularities in their pre-training data, such as the frequency of particular option labels or the distribution of answers appearing at particular positions (Zhao et al., 2021). IT models are additionally trained to follow semantic instructions, which reduces sensitivity to these surface features and grounds outputs more directly in the semantic content of the question (Ouyang et al., 2022).

While our evaluation is explicitly scoped to models in the 7B-9B parameter class, prior literature suggests that simply scaling up parameter counts does not automatically resolve format sensitivity. Elazar et al. (2021) demonstrate that while larger models consistently achieve higher accuracy, semantic inconsistencies under paraphrasing remain a pervasive issue regardless of scale. Similarly, Ruprecht et al. (2025) note that while higher-capacity models exhibit marginally greater stability on survey items, no model size proves inherently robust to scale or phrasing perturbations.

Finally, while we lack human baselines for these specific reworded and shuffled variants, typical human parallel-forms reliability for well-validated Likert scales generally produces Pearson correlations in the range of $r \approx 0.70$ to 0.85 (Lord et al., 1968; Nunnally, 1978). By this benchmark, our Instruction-Tuned models ($r \approx 0.70$-0.92) operate at or slightly above the threshold of typical human reliability, whereas Base models ($r \approx 0.30$-0.60) fail it completely. This contextualizes the performance of IT models not just as a mechanical improvement, but as achieving a baseline level of consistency comparable to human survey respondents.

5.2 Symbol Binding, Labels, and Prompt Design

The pattern from Section 4.3, where removing labels helps once the options are shuffled, points to a subtle but practically important difference in how instruction-tuned models handle labeled versus unlabeled prompts. Labels do two different jobs depending on the model. For base models, they are essential anchors that connect an abstract output token to a specific answer option; take them away, and that connection breaks down. For instruction-tuned models, though, labels introduce a second signal that can pull against the model's actual con-

tent preference, especially once the options have been shuffled. This tension between choosing by label and choosing by content fits with the symbol-binding failures documented by Xue et al. (2024), but here the fix turns out to be removing the label entirely rather than improving how it is bound to content.

There is likely also an alignment-training angle here. RLHF may leave models with a mild habit of favoring certain label tokens ("A" or "1") as a kind of learned response style, separate from what the content actually says. When the options are shuffled, that habit works against the model's real semantic preference and shows up as inconsistency. Drop the labels, and the habit has nothing to attach to. Some of the stability we see in instruction-tuned models is probably genuine semantic reasoning, and some of it may be a form of learned compliance (Sharma et al., 2024), but either way, our results show that the net effect of alignment training on survey reliability is clearly positive across the board.

6 Conclusion

We set out to test whether LLMs meet the parallel-forms reliability standard expected of survey instruments, by systematically varying Likert-scale wording, cardinality, polarity, and option order. Our main finding is straightforward: reliability in LLM-based survey research is not something 7-9B models come with out of the box, it is produced by instruction tuning and remains sensitive to how prompt labels are formatted. Base models fail basic reliability checks and should not be used as synthetic survey respondents. Instruction-tuned models cut response variance by roughly half and show meaningful structural robustness, but some scale sensitivity remains even for them. We also find a less obvious interaction between prompt format and option shuffling, where removing labels can improve reliability once option order is randomized, which has direct implications for how synthetic surveys should be designed. Future work should extend this evaluation to paraphrasing the question stems themselves, to larger models, to non-English survey corpora, and, most importantly, should compare LLM reliability against human parallel-forms reliability coefficients to see whether current instruction-tuned models meet the standards we apply to human respondents.

Limitations

Our evaluation is subject to several limitations. First, we rely on strict output parsing, which may conflate formatting failures with genuine reasoning failures (Molfese et al., 2025). Penalizing a model for selecting "Don't Know" in P4, for instance, ignores that expressing uncertainty can reflect appropriate calibration rather than inconsistency (Kadavath et al., 2022). Second, Mistral-7B-v0.3's high unusable rate in P4 means that the consistency metrics for that model are computed over a filtered, potentially non-representative subset of questions. Third, all inference uses greedy decoding; robustness under stochastic sampling and higher temperatures remains unexplored. Fourth, the study is limited to English-language questions and four 7-9B parameter models; consistency patterns may differ for larger models or non-English corpora. Fifth, and most critically, we lack human baselines answering the same reworded and shuffled variants, making it impossible to determine whether current LLM consistency levels are subhuman, comparable to human reliability, or in some cases superior ("hyper-accurate") but brittle in qualitatively different ways (Aher et al., 2023). Future work should also extend perturbations to question-stem paraphrasing (Elazar et al., 2021) and to a broader range of scale cardinalities and polarities (Menold, 2021).

Ethical Considerations

We use the English OpinionQA dataset (Santurkar et al., 2023), which is freely available for research. Our extended dataset is released exclusively for non-commercial research purposes and respects all intellectual property and privacy requirements of the original dataset. Experimental compute was performed on institutional GPU infrastructure. No personal information was collected. The authors acknowledge the use of ChatGPT solely for grammar correction and text coherence, and for assistance with coding.

Acknowledgments

Matthias Aßenmacher received funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the National Research Data Infrastructure-NFDI 27/1-460037581-BERD@NFDI.

References

Gati V Aher, Rosa I. Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 337–371. PMLR.

Arthur Couch and Kenneth Keniston. 1960. Yeasayers and naysayers: Agreeing response set as a personality variable. *Journal of Abnormal and Social Psychology*, 60(2):151–174.

Lee J. Cronbach. 1951. Coefficient alpha and the internal structure of tests. *Psychometrika*, 16(3):297–334.

Jennifer Dykema, Nora Cate Schaeffer, Dana Garbarski, Nadia Assad, and Steven Blixt. 2022. Towards a reconsideration of the use of agree-disagree questions in measuring subjective evaluations. *Research in Social and Administrative Pharmacy*, 18(2):2335–2344.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jan Karem Höhne, Dagmar Krebs, and Steffen M Kühnel. 2022. Measuring income (in)equality: Comparing survey questions with unipolar and bipolar scales in a probability-based online panel. *Social Science Computer Review*, 40(5):108–123.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Ariba Khan, Stephen Casper, and Dylan Hadfield-Menell. 2025. Randomness, not representation: The unreliability of evaluating cultural alignment in llms. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, page 2151–2165, New York, NY, USA. Association for Computing Machinery.

Jon A Krosnick. 1991. Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied Cognitive Psychology*, 5(3):213–236.

Jon A. Krosnick and Duane F. Alwin. 1987. An evaluation of a cognitive theory of response-order effects in survey measurement. *Public Opinion Quarterly*, 51(2):201–219.

Noah Lee, Jiwoo Hong, and James Thorne. 2025. Evaluating the consistency of LLM evaluators. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10650–10659, Abu Dhabi, UAE. Association for Computational Linguistics.

Frederic M. Lord, Melvin R. Novick, and Allan Birnbaum. 1968. *Statistical Theories of Mental Test Scores*. Addison-Wesley.

Riccardo Lunardi, Vincenzo Della Mea, Stefano Mizzaro, and Kevin Roitero. 2025. On robustness and reliability of benchmark-based evaluation of llms. In *Proceedings of the 28th European Conference on Artificial Intelligence (ECAI 2025)*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, pages 4603–4610. IOS Press.

Natalja Menold. 2021. Response bias and reliability in verbal agreement rating scales: Does polarity and verbalization of the middle category matter? *Social Science Computer Review*, 39(1):130–147.

Francesco Maria Molfese, Luca Moroni, Luca Gioffré, Alessandro Scirè, Simone Conia, and Roberto Navigli. 2025. Right answer, wrong score: Uncovering the inconsistencies of LLM evaluation in multiple-choice question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18477–18494, Vienna, Austria. Association for Computational Linguistics.

Lilian Ngweta, Kiran Kate, Jason Tsay, and Yara Rizk. 2025. Towards LLMs robustness to changes in prompt format styles. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 529–537, Albuquerque, USA. Association for Computational Linguistics.

Jum C. Nunnally. 1978. *Psychometric Theory*, 2nd edition. McGraw-Hill.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022.

Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 27730–27744, Red Hook, NY, USA. Curran Associates Inc.

Pew Research Center. 2024. The American Trends Panel (ATP). Dataset. Accessed via OpinionQA.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Jens Rupprecht, Georg Ahnert, and Markus Strohmaier. 2025. Prompt perturbations reveal human-like biases in large language model survey responses. *Preprint*, arXiv:2507.07188.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 29971–30004. PMLR.

Mrinank Sharma, Meg Tong, Tomek Korbak, David Duvenaud, Amanda Askell, Sam Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2024. Towards understanding sycophancy in language models. In *International Conference on Learning Representations*, volume 2024, pages 110–144.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Lindia Tjuatja, Valerie Chen, Tongshuang Wu, Ameet Talwalkwar, and Graham Neubig. 2024. Do LLMs exhibit human-like response biases? a case study in survey design. *Transactions of the Association for Computational Linguistics*, 12:1011–1026.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. 2024a. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9440–9450, Bangkok, Thailand. Association for Computational Linguistics.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024b. "my answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7407–7416, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mengge Xue, Zhenyu Hu, Liqun Liu, Kuo Liao, Shuang Li, Honglin Han, Meng Zhao, and Chengguo Yin. 2024. Strengthened symbol binding makes large language models reliable multiple-choice selectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4331–4344, Bangkok, Thailand. Association for Computational Linguistics.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations*, volume 2024, pages 19426–19454.

A Implementation Details

A.1 Data Preparation and Filtering

Starting from the 1,498 OpinionQA questions, a two-stage manual review reduced the dataset to 1,235 items suitable for scale-variation experiments. In the first stage, questions were excluded if they implicitly assumed a specific respondent background (e.g., "While growing up, how often did you . . . "; "How much . . . would you say is in your neighborhood"), because such items would require the model to adopt a demographic persona rather than express a generalized opinion. In the second stage, any remaining question whose options were structured as substantive statements rather than scalar degrees (e.g., "Which of the following statements comes closer to your view?" with two distinct policy positions) was reformulated into a Likert-administrable question. This involved converting the original stem into a statement and prefacing it with "How much do you agree or disagree with the following statement?", then treating each original option as a separate item. This adaptation preserved the substantive content of the original survey while enforcing a uniform Likert-scale structure.

Content dimension classification was performed with an automated Python script combining keyword matching and Regular Expressions. Question stems were matched against 15 dimension-specific keyword lists (e.g., stems containing *agree/disagree/support/oppose* were assigned to the "Agreement" dimension; those containing *how likely* or *probability* to "Likelihood"; those with comparative terms like *better/worse* to "Better/Worse"). The resulting distribution is shown in Table 4. Each question type was then paired with a set of five semantically equivalent but differently worded answer option sets. The Agree-Disagree (AD) format was adopted as the primary scale structure because it allows a uniform scale design across diverse question topics, isolating the effect of scale parameters (cardinality, phrasing, polarity) from question semantics, in line with the goal of studying instrument effects rather than item content effects. We acknowledge that AD scales impose an additional cognitive mapping step compared to item-specific scales (Dykema et al., 2022), but this limitation is secondary to the experimental goal of systematic scale variation.

For the Shuffled Dataset, five random permutations of the original option order were generated for each question using fixed random seeds (2, 7, 10, 67, 101), yielding six orderings per item (original plus five shuffled). The Shuffled Dataset uses scales of up to five options, as the factorial growth of 6-option permutations ($6! = 720$) makes systematic sampling impractical within the experimental budget.

A.2 Hardware and Decoding

All experiments were executed on a single NVIDIA A100 (80 GB) GPU, using the Hugging Face `transformers` library (Wolf et al., 2020). Models were loaded in native precision (`dtype="auto"`, typically bfloat16 or float16) to avoid performance degradation from quantization. Greedy decoding (`do_sample=False`) was used throughout, with `max_new_tokens=50`. Greedy decoding ensures that any observed variation across items is attributable solely to the changes in scale wording or option order, not to stochastic sampling.

A.3 Inference Pipelines

IT models received prompts via the model's native chat template in a single `user` turn (no system prompt). Omitting the system prompt was a deliberate choice: adding a persona-shaping system message (e.g., "You are a helpful assistant") would introduce an uncontrolled variable across model families, which implement system-prompt parsing differently. Base models received a raw text-completion prompt ending with a forcing suffix (e.g., "The best answer is option number:"). Without this suffix, Base models tend to continue the list of options as if completing a document, rather than selecting one. Model-specific end-of-sequence tokens were added to the stopping criteria to prevent the model from generating additional text after the answer token.

A.4 Content Dimensions

Table 4 shows the distribution of questions across the 15 content dimensions after filtering and adaptation. The "Agreement" dimension is the most prevalent (32.7%), reflecting the design of the original Pew Research surveys, which frequently ask respondents to express their degree of agreement with policy positions or factual statements. The next-largest dimension is "Quantity" (17.3%), capturing items about the appropriate amount or level of something (e.g., government spending, personal frequency of behavior). The remaining 13 dimensions each account for under 9% of the dataset,

providing a diverse coverage of opinion-related response formats while keeping the classification scheme manageable.

Content Dimension	Count	%
Agreement	404	32.7
Quantity	214	17.3
Better / Worse	101	8.2
Importance	91	7.4
Likelihood	85	6.9
Positive / Negative	67	5.4
Good / Bad	50	4.0
Magnitude of Problem	48	3.9
Frequency	38	3.1
Performance ("How Well")	38	3.1
Priority	35	2.8
Acceptance	31	2.5
Reason	15	1.2
Concern	11	0.9
Increase / Decrease	7	0.6
Total	**1,235**	**100.0**

Table 4: Content dimension distribution ($N = 1{,}235$).

B Prompt Templates

Design Rationale. The prompting strategy pursues two goals simultaneously: reliably extracting a single answer token, and minimizing extraneous biases not under study. The opening framing "You are taking a survey. There are no right or wrong answers" serves a dual purpose. It signals that the task is opinion-based rather than factual, reducing the likelihood of content-moderation refusals on politically sensitive items, and it suppresses the tendency of instruction-tuned models to hedge or justify rather than commit to a scalar response.

A **zero-shot** approach was adopted throughout. Including few-shot demonstration examples would introduce an additional positional confound: the label positions in the example items could prime the model toward particular options (e.g., a demonstration always selecting option A could bias subsequent selections). Zero-shot evaluation ensures that any systematic option preference reflects the model's own prior rather than a learned heuristic from the context.

For **IT models**, all content was delivered in a single `user` turn without a system prompt, as described in Appendix A. The output format constraint ("Provide exactly one [label] only") was added after pilot experiments revealed that instruction-tuned models frequently returned verbose explanations or cited multiple options unless explicitly instructed to output a single token.

For **Base models**, the instruction block was removed and replaced by a forcing suffix ("The best answer is option [label]:") appended directly after the option list. This text-completion framing exploits the base model's training objective: it predicts the next token following the suffix, which is typically the answer label.

The four prompt format variants (P1–P4) differ only in how options are labeled and presented. **P1 (Numeric)** uses "1) . . . 2) . . . " enumeration; **P2 (Alphabetic)** uses "A) . . . B) . . . "; **P3 (Unlabeled)** lists options as plain newline-separated text without any prefix; **P4 (Extended)** augments P1 with two additional options ("Don't Know" / "Refused") labeled $N{+}1$ and $N{+}2$, mapped to exclusion codes during analysis. For P3, the output-format instruction for IT models is abbreviated to "Please answer with ONLY one of the answer options given above" without specifying a label type, since no labels exist. Base models receive no forcing suffix under P3. The detailed template is shown in Figure 5.

C Supplementary Results

C.1 Dataset Scale Characteristics

Table 5 shows the distribution of scale types across the two experimental datasets. The Reworded Dataset includes 6-point scales (bipolar and unipolar) that do not appear in the Shuffled Dataset. This asymmetry arises because 6-point scales generate 720 possible permutations (6!), making systematic random sampling computationally and analytically impractical. The five seeds used for the Shuffled Dataset are instead applied to scales of up to five options, which yield 120 permutations at most. 4-point scales dominate both datasets (58.6% of Reworded, 78.7% of Shuffled) because the majority of content dimensions are served by a standard 4-point bipolar or unipolar structure. The 5-point bipolar format accounts for the next largest share, capturing dimensions (such as Importance or Likelihood) for which a neutral midpoint is semantically appropriate. The co-occurrence of variable cardinalities in the Reworded Dataset is a key driver of the smoother APD distributions described in Appendix C.7: when a 4-point and a 5-point variant are compared, the normalized distance values can take on more intermediate steps than when comparing two variants of identical length.

IT Models (Chat Template)	Base Models (Text Completion)

`You are taking a survey. There are no right or wrong answers.`
`Please choose the one option that best fits your view. The order of the options does not imply any ranking.`
`Output format: Provide exactly one [label] only (no words, no punctuation, no quotes).`
 	
`Question: [question]`
 	
`Options:`
`[options]`
 	
`Please answer with ONLY one of the answer options given above, and with only the [label] corresponding to the option you choose.`
	`Question: [question]`
	
	`Options:`
	
	`[options]`
	
	`The best answer is option [label]:`

Figure 5: Core prompt templates. `[label]` is substituted with "number" (P1), "letter" (P2), or "option" (P3). For P3 (Unlabeled), the IT prompt omits the output-format line and truncates at "*...given above.*"; no forcing suffix is used for Base models under P3. For P4, the option block is extended with numbered "Don't Know" and "Refused" entries.

	Reworded		Shuffled	
Scale Type	**N**	**%**	**N**	**%**
4-pt Unipolar	1,858	30.1	3,408	46.0
4-pt Bipolar	1,760	28.5	2,424	32.7
5-pt Bipolar	1,375	22.3	1,350	18.2
5-pt Unipolar	635	10.3	228	3.1
6-pt Bipolar	512	8.3	–	–
6-pt Unipolar	35	0.6	–	–

Table 5: Scale type distributions across the two datasets. P4 extends all scales by +2 options ("Don't Know" / "Refused"), not reflected in these counts.

C.2 IT Model Response Distributions

Figure 6 shows the response category distributions for IT models on the Reworded Dataset, and should be read alongside Figure 2 in the main paper, which covers Base models. The contrast is stark along two dimensions.

First, the **unusable rate** for IT models is negligible ($< 0.1\%$) across all four prompt formats, whereas Base models exhibit unusable rates ranging from around 8% under standard formats to nearly 35% under P4 for Mistral. This confirms that instruction tuning reliably enforces output format constraints: IT models correctly interpret the label-selection instruction and respond with a single token, even under the more complex P3 (Unlabeled) and P4 (Extended) formats that confuse base models.

Second, IT models display markedly **stable semantic profiles** across P1–P4. Unlike Base models, which shift their modal preference dramatically depending on whether options are labeled numerically or alphabetically (e.g., Gemma-2-9B selects Option 4 most often under P1 but shifts to a sharp spike at Option 1 under P2), IT models maintain broadly consistent response distributions regardless of the label type used. This structural stability is the behavioral correlate of the lower MSD values reported in Table 2 of the main paper, and suggests that instruction tuning reduces sensitivity to surface-level label format rather than merely suppressing invalid outputs.

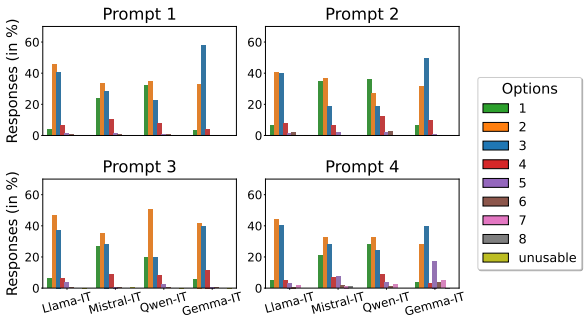


Figure 6: Response category distributions for IT models on the Reworded Dataset. Unusable rates are negligible (<0.1%) and response profiles are consistent across all four prompt formats, in sharp contrast to the Base model distributions in Figure 2.

C.3 Model-Specific Behavior Profiles

Beyond the general Base/IT contrast, our results reveal instructive model-specific patterns that illuminate sources of instability. **Gemma-2-9B (Base)** represents the most dramatic case of format-driven instability. Under Prompt 1 (Numeric), responses concentrate at the negative extreme (-1.0) on the Reworded Dataset, but flip to the positive extreme ($+1.0$) when shuffled. This polarity inversion suggests a strong position bias when numerical labels no longer reliably identify options. After instruction tuning, Gemma-2-9B-IT becomes the most consistent model, confirming that alignment training effectively resolves this sensitivity.

Qwen2.5-7B (Base) exhibits atypically high intrinsic stability, with inter-prompt correlations ($r = 0.849$) approaching IT-level performance. Its aggregate distributions remain similar between Reworded and Shuffled conditions, likely reflecting high-density structured MCQA reasoning in pre-training. **Mistral-7B-v0.3 (Base)** is the least structurally consistent Base model, with inter-prompt correlations frequently below 0.25. Its IT variant shows improvement but remains the least consistent IT model, suggesting alignment training is less effective at resolving underlying symbol-binding tensions.

C.4 The Qwen2.5 Base Model Anomaly

Qwen2.5-7B (Base) presents an atypical pattern among Base models. While other Base models show large distributional divergences between Reworded and Shuffled conditions, Qwen2.5-7B maintains relatively overlapping distributions across both datasets. Its inter-prompt correlations on the Reworded Dataset reach P1–P4: $r = 0.849$ and P1–P3: $r = 0.575$, values that are comparable to those of some IT models from other families (cf. Table 7) and substantially higher than the other Base models (e.g., Mistral Base P1–P3: $r = 0.408$, Gemma Base P1–P3: $r = 0.374$). Its MSD on the Reworded Dataset (P1: 0.254, lowest among all Base models) and the Shuffled Dataset (P1: 0.323, again lowest) further confirm this outlier status.

We attribute this stability to a high density of structured MCQA content in Qwen's pre-training data. Alibaba Cloud's Qwen series is trained on a corpus with a reportedly large fraction of question-answering and instruction-style data (Qwen et al., 2025), which may pre-train a rudimentary form of symbol binding even without explicit alignment. Figures 7 and 8 illustrate this: while other Base models show the blue (Reworded) and orange (Shuffled) projected-response distributions drifting substantially apart (indicating that option shuffling changes the apparent semantic opinion), Qwen's distributions overlap substantially across all four prompt formats.

The IT variant of Qwen further smooths and narrows the distributions but provides a smaller absolute improvement than seen in other model families. For instance, Llama's MSD drops by roughly 42% from Base to IT on P1 ($0.357 \rightarrow 0.207$), whereas Qwen's improvement is only around 19% ($0.254 \rightarrow 0.232$), suggesting diminishing marginal returns of alignment for a model that already exhibits partial stability at the pre-training stage.

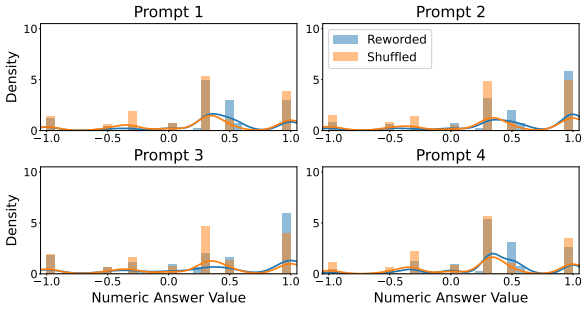


Figure 7: Projected response distributions for Qwen2.5-7B (Base). Blue: Reworded Dataset; Orange: Shuffled Dataset. The two distributions overlap substantially across all four prompt formats, unlike most Base models, which diverge sharply when options are shuffled.

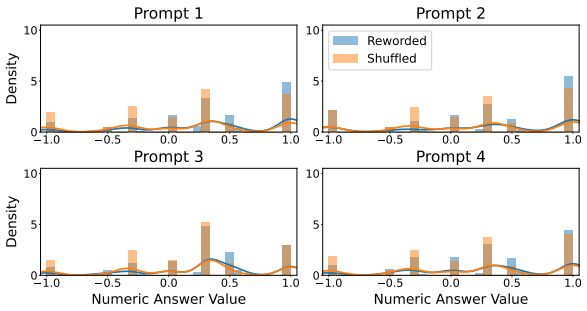


Figure 8: Projected response distributions for Qwen2.5-7B-Instruct (IT). Alignment further narrows the distributional spread, but the high baseline overlap observed in the Base model means the marginal gain from instruction tuning is smaller than for other families.

C.5 MSD on the Shuffled Dataset

Table 6 reports the MSD on the Shuffled Dataset and provides the complete quantitative basis for the Label-Free Consistency Paradox discussed in

Section 4.3 of the main paper.

For IT models, the P3 (Unlabeled) column is bolded because it achieves the lowest MSD for every model family. The reductions are substantial: Gemma-2-9B-IT drops from 0.232 (P1) to 0.121 (P3), a reduction of 47.8%; Llama-3.1-8B-IT drops from 0.277 (P1) to 0.146 (P3), a reduction of 47.3%; Qwen2.5-7B-IT drops from 0.310 (P1) to 0.206 (P3), a reduction of 33.5%. Even Mistral, which shows elevated P3 variance in the Reworded Dataset (Table 2), achieves its second-lowest MSD under P3 in the Shuffled condition (0.240 vs. 0.263 for P1).

The mechanism is as follows. Under labeled formats (P1, P2, P4), shuffling the options creates a conflict between the model's positional prior (e.g., a tendency to select option "A" or "1") and its semantic preference (e.g., a tendency to select content meaning "Agree"). Because these two cues point to different options after shuffling, the model's outputs become inconsistent across orderings. Under P3, no labels are present, so the positional prior is over raw text position rather than a symbol. Since IT models are better calibrated to attend to content, the removal of labels forces them to rely more directly on semantic information, which is invariant across orderings. The result is higher consistency under the very condition that most disrupts labeled formats.

For Base models, this paradox does not materialize systematically. Base model P3 values show no consistent pattern: Llama P3 (0.445) is the highest across its prompts, whereas Mistral P3 (0.365) is the lowest. This is expected: Base models lack reliable symbol binding in the first place, so removing labels does not alleviate a label-induced conflict but instead eliminates the only structural cue that anchored their responses.

Model	P1	P2	P3	P4
Base Models				
Llama-3.1-8B	0.367	0.403	0.445	0.328
Mistral-7B-v0.3	0.503	0.400	0.365	0.482
Qwen2.5-7B	0.323	0.361	0.363	0.330
Gemma-2-9B	0.553	0.346	0.415	0.378
IT Models				
Llama-3.1-8B-IT	0.277	0.233	**0.146**	0.274
Mistral-7B-IT	0.263	0.320	**0.240**	0.249
Qwen2.5-7B-IT	0.310	0.277	**0.206**	0.273
Gemma-2-9B-IT	0.232	0.187	**0.121**	0.220

Table 6: MSD on the Shuffled Dataset. Bold P3 values for IT models confirm the Label-Free Consistency Paradox: removing option labels yields the lowest inter-ordering variance for all four IT model families, even though P3 is the most volatile format in the Reworded Dataset.

C.6 Full Correlation Matrix (Reworded Dataset)

Table 7 reports all six pairwise Pearson correlations between prompt formats on the Reworded Dataset. Several structural patterns are noteworthy.

First, **P1–P4 correlations are consistently the highest** across all model families, for both Base and IT variants (e.g., Mistral-IT: 0.913; Llama-IT: 0.894; Gemma-IT: 0.897). This reflects the structural similarity between these two formats: P4 is a direct extension of P1 (numeric labels) with two additional options appended, so models that have converged on a numeric-label strategy under P1 apply essentially the same logic under P4, differing only in whether they ever select the "Don't Know" or "Refused" options.

Second, **pairs involving P3 (Unlabeled) tend to produce lower correlations**, particularly P1–P3 and P3–P4. For IT models, P1–P3 correlations range from 0.701 (Llama) to 0.778 (Gemma), compared to P1–P4 values of 0.894–0.897. This reflects the fundamentally different structural demands of P3, which requires the model to parse and select from plain-text options rather than binding a numeric or alphabetic symbol to content. The label-free format creates a distributional shift in response tendencies (visible in the IT MSD data of Appendix C.5), which lowers correlation with labeled formats.

Third, all correlations in Table 7 are substantially higher than their counterparts in the Shuffled correlation matrix (Table 3 in the main paper). This confirms at the level of individual-question response profiles that semantic rewording is a milder perturbation than positional shuffling: models preserve the relative direction of their opinions across reworded variants better than across shuffled orderings. For IT models, many Reworded correlations exceed 0.80, whereas Shuffled correlations rarely approach this range.

C.7 APD Distributions: Reworded Dataset

Figures 9 and 10 show APD distributions on the Reworded Dataset under P1, and complement the Shuffled Dataset distributions in the main paper.

The most salient structural difference between the two datasets is the shape of the distributions. The **Shuffled Dataset produces discrete spikes**

Model	1–2	1–3	1–4	2–3	2–4	3–4
Base						
Llama-3.1	0.580	0.398	0.644	0.464	0.536	0.461
Mistral-7B	0.485	0.408	0.454	0.354	0.336	0.278
Qwen2.5-7B	0.608	0.575	0.750	0.467	0.560	0.481
Gemma-2-9B	0.534	0.374	0.564	0.473	0.627	0.504
IT						
Llama-3.1	0.820	0.701	0.894	0.712	0.797	0.683
Mistral-7B	0.707	0.722	0.913	0.646	0.709	0.703
Qwen2.5-7B	0.818	0.743	0.883	0.737	0.799	0.726
Gemma-2-9B	0.860	0.778	0.897	0.820	0.841	0.749

Table 7: Pearson's r across all prompt-format pairs on the Reworded Dataset. All values exceed their Shuffled counterparts (Table 3), confirming that semantic rewording is less disruptive than positional permutation. P1–P4 pairs are consistently the highest due to structural similarity between numeric-label and extended-numeric formats; pairs involving P3 (Unlabeled) are consistently lower.

at a small set of APD values. This is a direct mathematical consequence of fixed scale cardinality: when all five or six variants of a question share the same scale length, the normalized values at which responses can land are constrained to a finite set (e.g., for a 4-point scale mapped to $\{-1.0, -0.33, 0.33, 1.0\}$, the only possible pairwise distances are 0, 0.67, 1.33, and 2.0). As a result, the APD for each question takes on one of a small number of discrete values, creating visible spikes in the density plot at those locations.

The **Reworded Dataset produces smoother, more continuous distributions**. Because the five reworded variants for a given question may draw from scales of different cardinalities (e.g., a 4-point variant paired with a 5-point variant), the set of possible normalized distances is richer. Cross-cardinality pairings generate intermediate values that do not appear in any single-cardinality comparison, effectively densifying the support of the APD distribution and smoothing out the discrete spikes.

For Base models (Figure 9), the distributions are wide and exhibit substantial density in the high-APD region, with a non-trivial fraction of questions producing APDs above 1.0, indicating near-maximal scale traversals triggered by mere wording changes. For IT models (Figure 10), the distributions are strongly compressed toward 0, with most questions achieving APD values below 0.5. The smooth shape is preserved because the variable scale cardinalities apply equally to Base and IT models; the compression reflects reduced sensitivity to wording rather than any change in scale structure.

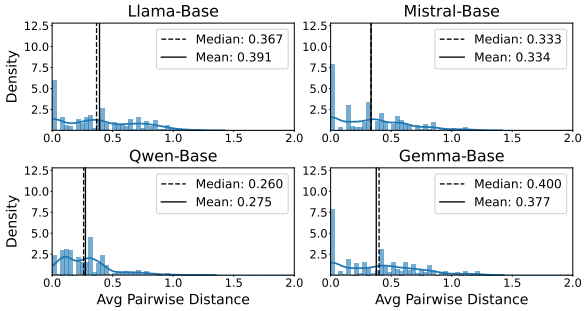


Figure 9: APD distributions for Base models on the Reworded Dataset (P1). The smooth, continuous shape arises from variable scale cardinalities across reworded variants. Wide dispersion and substantial density above APD $= 1.0$ indicate that wording changes alone can induce near-complete polarity reversals in Base models.

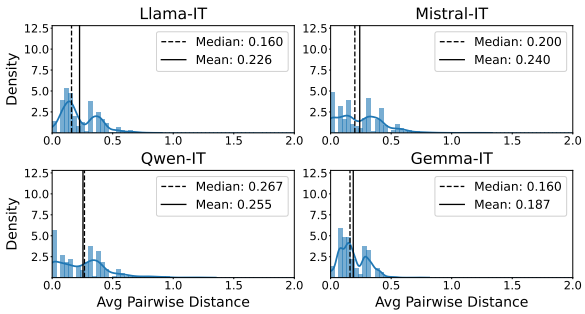


Figure 10: APD distributions for IT models on the Reworded Dataset (P1). The smooth curve shape is maintained by the same variable-cardinality structure, but distributions are strongly compressed toward 0 relative to Base models, reflecting substantially reduced sensitivity to semantic rewording.

Evaluating Token Probabilities as Confidence Estimators in Human Behavior Simulations: A Case Study in Luxembourgish Next Like Prediction

Nils Schwager[1], **Simon Münker**[1], **Alistair Plum**[2], **Achim Rettinger**[1]
[1]Trier University, Trier, Germany
[2]University of Luxembourg, Esch-sur-Alzette, Luxembourg
{schwager, muenker, rettinger}@uni-trier.de
alistair.plum@uni.lu

Abstract

As large language models (LLMs) are increasingly used to simulate human behavior, their operational validity is typically assessed through accuracy alone, implicitly treating human behavior as deterministic. We argue that on noisy digital trace data, a model's performance depends not only on predicting the correct outcome but on its calibration. We present a case study testing whether the token probabilities of open-weight LLMs can serve as confidence estimates in a Next Like Prediction task: predicting how a user of the Luxembourgish news platform RTL.lu rates a comment, conditioned on their prior interactions. Evaluating Ministral 3 (3B, 8B), we find that, with in-context learning only, the tested models are overconfident and are outperformed on accuracy and calibration by a simple statistical baseline. Accuracy and calibration do not move together: scaling from 3B to 8B improves accuracy while degrading calibration, and longer behavioral histories raise accuracy while confidence declines. Task-specific supervised fine-tuning acts as an effective calibration intervention, reducing Expected Calibration Error from 0.2 to 0.006 while improving accuracy from 0.536 to 0.758. We conclude that confidence estimation should be evaluated alongside accuracy to improve validity when using LLM-based agents.

1 Introduction

As large language models (LLMs) are increasingly used to simulate human behavior, methodological research into the validity (Larooij and Törnberg, 2025; Münker et al., 2026b) and alignment of generative agents (Chuang et al., 2024; Schwager et al., 2026) emphasizes that achieving operational validity requires alignment with authentic digital trace data. However, this type of data is inherently noisy due to the non-deterministic nature of human behavior and missing contextual modalities (Ouellette and Wood, 1998; Zhang et al., 2021). This lack of information makes it difficult to achieve sufficient predictive performance, or even to determine confidence thresholds.

Contemporary research utilizing LLMs as generative agents mostly relies on evaluating alignment using discrete (Lu et al., 2026; Wang et al., 2026) or continuous accuracy measures (Shen et al., 2025b; Schwager et al., 2026), thereby treating human behavior as deterministic. We argue that in simulations (and LLM applications processing noisy and/or user-generated text in general), the LLM's utility depends not merely on its ability to predict the correct outcome, but on its capacity to accurately estimate its own predictive confidence (Farquhar et al., 2024; Aichberger et al., 2025; Nguyen et al., 2026). Because we will never reach absolute certainty on noisy data, we see calibration[1] as a requirement for model alignment that is as important as accuracy itself (Azaria and Mitchell, 2023; Bakman et al., 2024).

To address this gap, we present a case study investigating whether LLMs' token probabilities can natively serve as confidence metrics in human behavior simulation, and we evaluate task-specific fine-tuning as a calibration intervention. We utilize a **Next Like Prediction** task based on user comments and interactions from the Luxembourgish news platform RTL.lu[2]. Our task tests whether an **LLM can accurately predict a user's like or dislike of a given comment, based solely on previous (dis)likes.** The setup isolates calibration for two reasons: (A) it introduces severe linguistic complexity and contextual sparsity (Section 3.3), and (B) it collapses human feedback into a strictly binary action space (like vs. dislike), which simpli-

[1]We define *calibration* as the degree of alignment between a model's confidence and its accuracy. Conversely, we use *calibration intervention* to refer to the methodological procedure to align these probabilities with empirical observations - in our case, task-specific fine-tuning.

[2]RTL Lëtzebuerg: https://www.rtl.lu/

Proceedings of the 11th Workshop on Natural User-generated Text, pages 89–100
October 28, 2026 ©2026 Association for Computational Linguistics

fies the measurement of accuracy and confidence relative to open-text alignment.

1.1 Research Questions

We investigate two research questions:

RQ_1 How well do LLMs predict user likes/dislikes on Luxembourgish comments, and to what extent can we rely on their token probabilities as estimated confidence of these predictions?

RQ_2 How does task-specific fine-tuning impact both the discrete accuracy and the underlying calibration of the model's token probabilities?

1.2 Contributions

Our work makes the following contributions to the evaluation of LLM-based user modeling:

Confidence as an Evaluation Axis We argue that alignment for generative agents should be assessed through calibration alongside accuracy, and we operationalize this by treating the normalized token probabilities of a binary decision as a native confidence estimate for a human behavior simulation task.

Accuracy and Calibration Divergence Using a Next Like Prediction task on Luxembourgish digital trace data, we show that both tested off-the-shelf models are overconfident: they assign high probabilities to incorrect predictions and are outperformed on both accuracy and expected calibration error by a simple statistical baseline. Crucially, the two signals diverge: scaling from 3B to 8B improves accuracy while increasing calibration error, and lengthening the behavioral history improves accuracy while confidence declines. Especially in the first case, accuracy alone would misreport the model's probabilistic alignment.

Fine-Tuning as a Calibration Intervention We show that task-specific supervised fine-tuning substantially reduces calibration error and widens the gap between confidence on correct and incorrect predictions, indicating that lightweight adaptation can bring a model's predictive probabilities into closer agreement with its observed accuracy.

2 Background

2.1 Luxembourgish NLP: Low Research but High Resource

Luxembourgish is the national language of the Grand Duchy of Luxembourg, and one of the four official languages alongside French, German, and German sign language. It is spoken natively by around 400,000 people (Fehlen et al., 2023) in a society marked by extensive multilingualism and code-switching. These conditions, a small speaker base, no single fully codified written standard, and constant contact with two much larger (in terms of number of speakers) neighbouring languages, have long made Luxembourgish a difficult target for NLP (Purschke, 2025).

A first strand of work on online discourse approached Luxembourgish through the lens of public attitudes towards multilingualism. Combining a Word2Vec based analysis of user comments from the RTL.lu news platform with crowdsourced questionnaire data collected through the Schnëssen project, Purschke (2020) connected sociolinguistic theories of language attitude with computational text analysis. A second strand similarly worked on the same online comments, but took an NLP approach in training various classifiers to moderate the comments (Ranasinghe et al., 2023). The work ultimately showed that multilingual models of the time worked well for this task, but lacked nuanced knowledge to effectively tackle problematic comments that should not be published.

In terms of language models for Luxembourgish, Philippy et al. (2024a) addressed the shortage of parallel data for Luxembourgish by compiling a small human generated cross-lingual parallel dataset, which was used to train LuxEmbedder, a sentence embedding model with cross-lingual capabilities. Lutgen et al. (2025) presented the first sequence to sequence normalisation models for Luxembourgish, built on the ByT5 and mT5 architectures and trained on word level data capturing real life spelling variation, in order to deal with the large amount of variation.

In addition to the growing model space for Luxembourgish (see Lothritz et al., 2022; Plum et al., 2025), the recent availability of benchmarks (see Philippy et al., 2024b; Plum et al., 2026) highlights an important aspect of the current situation that Luxembourgish finds itself in with relation to NLP. Although Joshi et al. (2020) classified Luxembourgish as "scraping-by", and current research is ongoing but limited, the availability of data appears more as a "high resource" situation. Many sources of data are growing and becoming more readily available, including news articles and user comments from the main news broadcaster of Luxembourg, an expanding Wikipedia, transcripts of

national assembly debates, online dictionary examples, to name but the largest collections. This difference in available data versus available methods highlights that processing and analysing the Luxembourgish language poses limiting difficulties, which are also apparent in the study at hand.

2.2 Calibration in Generative Agents

In machine learning, model calibration refers to the alignment of predictive probabilities (estimated confidence) with the observed accuracy (Geng et al., 2024; Guo et al., 2017).

Confidence in LLMs Applying traditional calibration methods directly to LLMs presents significant structural challenges. Unlike standard discriminative models, the output space for open-text generation grows exponentially with sequence length. While white-box methods attempt to estimate generation uncertainty using native logit scores, these indicators frequently suffer from surface form competition, where different valid textual representations compete for the same probability mass, skewing the actual confidence distribution (Holtzman et al., 2021). Consequently, researchers have explored alternative white-box approaches, such as probing internal state activations (Kadavath et al., 2022; Li et al., 2023). Black-box approaches require no access to the model's internal computational graph. Instead, these methods attempt to estimate uncertainty through consistency checks across multiple stochastic generations (Manakul et al., 2023), evaluations via surrogate models (Shrivastava et al., 2023), or by directly prompting the model to verbalize its uncertainty, yielding a "linguistic confidence" score (Mielke et al., 2022).

Generative Agents Recent methodological frameworks require the operational validity of generative agents to be established against authentic digital traces (Larooij and Törnberg, 2025; Schwager et al., 2026). We argue that for noisy, non-deterministic tasks such as simulating human behavior on digital trace data, quantifying calibration is a methodological requirement. Robust confidence estimation enables researchers to aggregate individual predictions and apply nuanced, probabilistic interpretations to computational social science simulations.

Research Gap This paper takes the probability a model attaches to a single predicted human action and checks it against whether that action occurred. Two lines of work approach that measurement and stop short. Calibration research assumes a resolvable ground truth, such as factual question answering or text labels with gold annotations (Gligorić et al., 2025; Tian et al., 2023), so that a prediction can be scored against the correct answer. Behavior simulation gives up the probability instead of the ground truth. The evaluation of generative agents so far rests on believability or, more recently, on accuracy metrics (Larooij and Törnberg, 2025). Neither asks whether a generative agent knows when it is wrong.

Human behavior offers no resolvable target: an observed action is one realization of a distribution over choices the individual could plausibly have made; like annotator disagreement on subjective labeling tasks, this variation is a property of the phenomenon rather than noise (Plank, 2022). Because that uncertainty cannot be removed from the task, we begin by moving the evaluation of generative agents from discrete accuracy to probability: accuracy reports a determinate result for an outcome that was never determinate, whereas calibration asks whether its confidence matches its accuracy.

We therefore test calibration in generative agents on the binary prediction of user actions (likes or dislikes) over authentic, non-English, code-switched digital trace data; the binary format allows direct extraction of token probabilities as native confidence markers.

3 Methods

3.1 Next Like Prediction Task

The Next Like Prediction task centers on predicting how a specific user would respond to a given stimulus, in this case, evaluating whether they would *like* (1) or *dislike* (0) a comment. By comparing predicted interactions with authentic digital traces, we assess whether models can capture individual-level behavioral patterns rather than producing generic responses. This framing strictly follows the operational validity criterion: alignment should be measured against the actual individuals being simulated, not abstract notions of plausibility (Larooij and Törnberg, 2025; Münker et al., 2026a).

Following the methodology established by Schwager et al. (2026), we rely on implicit conditioning by providing the model with a behavioral history. Specifically, we format previous user interactions as preceding prompt-completion turns in the LLM's native chat structure. To standardize

this conditioning and evaluate the impact of context scaling, the model receives a minimum of 3 historical interactions to predict the subsequent fourth interaction, with the context window expanding up to a maximum of 72 interactions. This effectively turns our prediction task into a classification task for a user's $(n+1)$-th interaction, based on the previous n interactions. During evaluation, we strictly condition the model on the user's ground-truth behavioral history, explicitly omitting the model's own prior predictions to prevent compounding errors.

In addition to the behavioral history, the model is conditioned with the following system prompt: *You are a social media user evaluating online posts. Keep your binary predictions (1 for like, 0 for dislike) consistent with your previous interaction history and the preferences you have expressed earlier.*

3.2 Dataset

The data for this task is provided by RTL.lu, the main news broadcaster of Luxembourg and the only one entirely in Luxembourgish. On the `rtl.lu` website, registered users can post comments below journalistic news articles. Users can react to the comments of other users with a *like* or a *dislike*. We are provided this data, which allows us to reconstruct which user liked or disliked which comments.[3] The oldest comment in the dataset is from March 2020 and the newest from May 2024.

Pre-Processing First, we reorganize reactions by user to enable user-level modeling. We model the users strictly as voters; the stimuli are comments made by other users. To prevent overfitting to power users, we remove the top 20% of users by interaction count. This leaves users with at most 72 stimulus–response pairs. Users with fewer than four interactions are excluded.

Splits & Size We partition data at the user level so that all stimulus–response pairs from a single user appear exclusively in training or testing. This prevents cross-user leakage and enables the evaluation of cross-user generalization. Our resulting training set comprises $3,247$ users and $57,185$ training cases (excluding the three required seed conditioning cases), as well as 783 test users and $13,759$ test cases.

[3] At the time of writing this article, we are unable to share the data due to privacy agreements. We hope that in the future, a privacy protected version may be made publicly available.

Descriptive Statistics The filtered dataset exhibits a baseline class distribution of 61% likes and 39% dislikes, establishing the base rate for the classification task. Within the defined bounds, users have an average of 20.5 interactions (median = 13). Furthermore, the user comments have an average length of 49 words. These comments are written in Luxembourgish (75%), English (15%), French (4%), and German (3%), reflecting the multilingual user base of the platform.

3.3 Task and Data Constraints

To contextualize the task difficulty, we note that the accuracy for this prediction task is inherently bounded by ambiguity in the digital trace data, stemming from three primary sources:

Linguistic Complexity The dataset contains mainly Luxembourgish, which also appears in mixed and unmixed combinations with German, French, and English. Furthermore, colloquial Luxembourgish exhibits significant variation and code-switching. This linguistic diversity creates a challenging setting, which also tests the models' different language capabilities.

Contextual Sparsity The prediction task relies exclusively on a sparse historical context window ranging from 3 to 72 previous interactions. Because the parent journalistic articles and broader external contexts are deliberately omitted from the input, the model must infer the initial topic solely from the target comment text. The model must subsequently derive a generalized user preference from this construct to execute the prediction.

Behavioral Non-Determinism Our data lacks information on latent variables that heavily influence behavior, such as offline events. Further, we assume human interaction is fundamentally non-deterministic; a user might upvote a comment on one day but ignore a similar one on another. Because these latent factors are collapsed into a binary outcome, perfect prediction is not expected (Hofman et al., 2017). Consequently, this task focuses on evaluating the model's ability to appropriately reflect its confidence given the noise.

3.4 Metrics

We use standard accuracy as a discrete measure and treat the token probabilities for the tokens 0 and 1 as the model's confidence. We acknowledge that interpreting token probabilities as confidence

is bounded by factors such as first-token deviations (Wang et al., 2024) and surface form competition (Holtzman et al., 2021). However, this limitation is mitigated in our setup because the model strictly predicts discrete states (0 or 1) and is explicitly conditioned to enforce format compliance (Section 3.1). We apply a standard softmax function ($t = 1$) over the full vocabulary and extract the probabilities for our target tokens (0 and 1). In a second step we normalize the probability mass of 0 and 1 to ensure they form a valid probability distribution over the binary target space.

Expected Calibration Error (ECE) To quantify the alignment between the model's confidence and empirical reality, we calculate the Top-1 ECE (Guo et al., 2017). We partition the normalized predictions into $M = 10$ equally spaced confidence bins. ECE measures the weighted average of the absolute difference between the empirical accuracy and the mean predicted confidence within each bin, thereby penalizing models that are systematically over- or underconfident.

Brier Score The Brier score evaluates the mean squared difference between the model's predicted probabilities and the actual binary outcomes (Brier, 1950). Unlike discrete accuracy, the Brier score penalizes extreme overconfidence on incorrect predictions.

ROC-AUC To measure the model's underlying discriminative power independent of a rigid 0.5 decision threshold, we report the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). This evaluates the probability that the model assigns a higher token probability to a randomly chosen positive interaction (like) than to a randomly chosen negative one (dislike).

Conditional Confidence To diagnose the calibration deficit, we disaggregate the model's certainty into two complementary metrics: Confidence on Correct Predictions ($\text{Conf}_{\text{Correct}}$) and Confidence on Incorrect Predictions ($\text{Conf}_{\text{Incorrect}}$). A model whose confidence is informative will exhibit a large gap between these values: high certainty when accurate, and appropriate uncertainty (probabilities approaching 0.5) when predicting incorrectly.

3.5 Experimental Setup

Model & Fine-Tuning We utilize the open-weight Ministral 3 model family (`Ministral-3-3B-Instruct-2512-BF16` and `Ministral-3-8B-Instruct-2512-BF16` (Liu et al., 2026)) because its robust pre-training on diverse European languages establishes a strong linguistic baseline for Luxembourgish text, while remaining parameter-efficient. We apply Supervised Fine-Tuning (SFT) to the models on the task described in Section 3.1. To ensure comparability across models, we use identical hyperparameters: one epoch, a maximum sequence length of $4,500$ tokens, and training on complete input sequences (system prompt, user prompts, and model completions). We use the paged AdamW optimizer with 8-bit quantization (Dettmers et al., 2022) to enable training on a single NVIDIA L40S GPU (48GB VRAM). All remaining hyperparameters follow the TRL defaults (von Werra et al., 2020).

Statistical Baseline To establish a non-semantic lower bound, we implement a naive statistical baseline. For each prediction step, the probability of a positive interaction is computed strictly as the distribution of the user's prior interactions (total past likes divided by total historical interactions).

Proprietary Baseline To establish a baseline of proprietary model capabilities, we evaluate `Gemini 3.5 Flash-Lite` (with reasoning disabled), a recent high-throughput model optimized for latency-sensitive classification tasks. As with most SOTA proprietary models, the internal logits remain inaccessible via the standard API, hence we only report its discrete predictive accuracy.

4 Results

4.1 Model Calibration (RQ_1)

To address RQ_1, we first evaluate the off-the-shelf models to observe how well they estimate their own certainty on noisy digital trace data. The results, detailed in Table 1 and visualized in the reliability diagram (Figure 1), show that these LLMs are overconfident in the tested setting. When evaluated on authentic user interactions, the 8B off-the-shelf model achieves a discrete accuracy of 0.5355, performing only slightly above random chance. Notably, this underperforms our simple statistical baseline, which achieves an accuracy of 0.5900 by strictly predicting the user's historical marginal distribution. Despite this low predictive accuracy, the 8B model exhibits a high ECE of 0.2004 and a Brier score of 0.2937, a penalty worse than simply predicting a constant 0.5 probability

Model	Accuracy (↑)	ROC-AUC (↑)	Brier Score (↓)	ECE (↓)	Conf Cor. (↑)	Conf Incor. (↓)
Statistical Baseline	0.5900	0.5488	0.2476	0.0642	0.6581	**0.6346**
Ministral 3 3B	0.4906	0.6128	0.2813	0.1650	0.6632	0.6482
Ministral 3 3B (SFT)	0.7089	0.7600	0.1880	0.0087	0.7303	0.6434
Ministral 3 8B	0.5355	0.5872	0.2937	0.2004	0.7494	0.7202
Ministral 3 8B (SFT)	**0.7578**	**0.8253**	**0.1618**	**0.0056**	**0.7828**	0.6616
Gemini 3.5 Flash Lite	0.6868	-	-	-	-	-

Table 1: **Comparative Model Performance and Calibration.** Results detail discrete accuracy and probabilistic calibration across all evaluated configurations. Best values per metric are **bolded**.

for every interaction.

This overconfidence becomes clear when looking at the conditional confidence metrics. When the 8B model makes an incorrect prediction, it still assigns a mean probability of 0.7202 ($\text{Conf}_{\text{Incorrect}}$) to its choice. This is nearly as high as its confidence when it is correct ($\text{Conf}_{\text{Correct}} = 0.7494$). In contrast, the purely statistical baseline naturally avoids this overconfidence, maintaining a much lower $\text{Conf}_{\text{Incorrect}}$ of 0.6346. The 3B model behaves similarly to the 8B, demonstrating a $\text{Conf}_{\text{Incorrect}}$ of 0.6482 against an overall accuracy of only 0.4906.

4.2 Fine-Tuning and Calibration Improvement (RQ_2)

To answer RQ_2, we examine the impact of task-specific SFT on the models' probability outputs. We find that fine-tuning improves the models' ability to appropriately express uncertainty while simultaneously increasing their discrete performance.

As shown in Table 1, fine-tuning the 8B model increases predictive accuracy to 0.7578 and raises ROC-AUC to 0.8253. This gain over the statistical baseline suggests that the model learns semantic behavioral patterns rather than defaulting to a user's historical average. The same intervention aligns the model's predicted probabilities more closely with its observed accuracy: fine-tuning reduces the ECE to 0.0056 and lowers the Brier score to 0.1618.

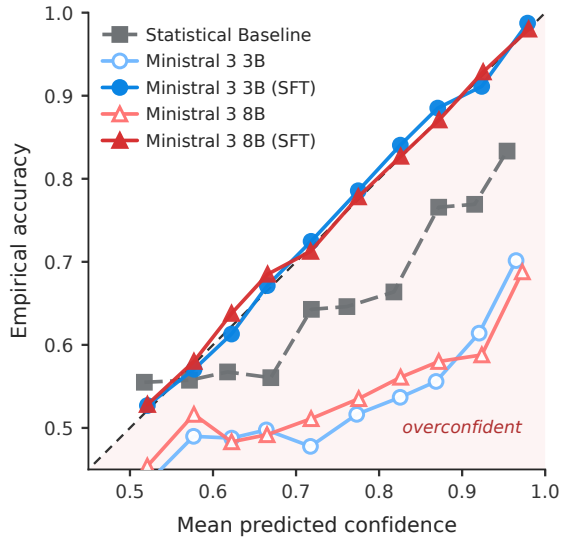


Figure 1: Reliability diagram comparing model calibration on the Next Like Prediction task.

This improvement is also visible in the conditional confidence gap. For the fine-tuned 8B model, mean confidence on correct predictions rises to 0.7828, while confidence on incorrect predictions falls to 0.6616. This widening gap is consistent with a model that lowers its probability estimates when its predictions are more likely to be wrong.

Finally, we observe that achieving good calibration does not strictly require larger models. The 3B (SFT) model achieves calibration nearly on par with the 8B (SFT) model (ECE 0.0087 vs. 0.0056), indicating that strong calibration is attainable at smaller scale, although the 8B (SFT) model remains best on both accuracy and calibration. Overall, task-specific fine-tuning proves to be effective not only for predicting user interactions but also for ensuring the model's confidence scores are sufficiently calibrated for downstream analysis.

When evaluated on the same dataset, the proprietary `Gemini 3.5 Flash-Lite` model achieves a discrete accuracy of 0.6868. While this outperforms the statistical baseline and off-the-shelf models, it falls short of our fine-tuned 8B model.

4.3 Contextual Scaling Analysis

As a supplementary analysis, we examine accuracy and confidence as a function of behavioral history length. The results, detailed in Table 2, reveal a divergence between accuracy and confidence in the off-the-shelf models. While providing extended behavioral history significantly improves the accuracy of the 8B model ($r = 0.50$), its internal confidence exhibits a strong inverse trajectory ($r = -0.64$). As the context window expands, the

Model	History → Acc		History → Conf	
	Δ_{100}	r	Δ_{100}	r
Statistical Baseline	0.195	0.49^{***}	-0.017	-0.20
Ministral 3 3B	0.100	0.23	-0.068	-0.71^{***}
Ministral 3 3B (SFT)	0.095	0.34^{**}	0.050	0.49^{***}
Ministral 3 8B	0.247	0.50^{***}	-0.087	-0.64^{***}
Ministral 3 8B (SFT)	0.086	0.33^{**}	0.045	0.43^{***}
Gemini 3.5 Flash Lite	0.143	0.43^{***}	-	-

Table 2: **Contextual Scaling.** Δ_{100} represents the linear change in accuracy and confidence per 100 interactions, rescaled from a slope fitted over the observed range (at most 72 interactions) and r denotes the Pearson correlation coefficient. Significance levels: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

model becomes more accurate but less confident.

Conversely, Supervised Fine-Tuning (SFT) realigns these metrics. For the fine-tuned 8B model, both accuracy ($r = 0.33$) and confidence ($r = 0.43$) scale positively and significantly with history length. This indicates that task-specific adaptation trains the model to utilize extended digital traces not just for discrete prediction, but to increase its confidence.

Finally, while the purely statistical baseline and the proprietary `Gemini 3.5 Flash-Lite` model both achieve significant accuracy gains with longer historical contexts ($r = 0.49$ and $r = 0.43$, respectively), their usable confidence signal is constrained. The baseline's confidence shows no significant association with history length ($r = -0.20$).

For the off-the-shelf models, this supports our central claim that accuracy alone is an insufficient evaluation criterion, since accuracy and the confidence signal move in opposite directions.

5 Discussion

5.1 Interpreting the Calibration Deficit

The results from Section 4 highlight a divergence between the models' discrete predictive capacity and their calibration. First, we observe that the statistical baseline natively avoids overconfidence because it anchors its predictions strictly to the distribution of historical interactions. This results in the lowest mean probability assigned to incorrect predictions (0.6346). In contrast, the off-the-shelf models exhibit a calibration deficit, assigning high probability mass to incorrect outcomes, which inflates their Brier scores.

Second, our findings indicate that accuracy and calibration do not inherently scale together. Increasing the parameter count from the 3B to the 8B model improves accuracy (from 0.4906 to 0.5355) but increases the ECE from 0.1650 to 0.2004. This decoupling demonstrates that parameter scaling alone does not resolve calibration deficits.

Third, SFT functions as an effective calibration intervention. Task-specific fine-tuning closes the gap between mean confidence and accuracy, and widens the gap between $\text{Conf}_{\text{Correct}}$ and $\text{Conf}_{\text{Incorrect}}$. Following SFT, the 8B model increases its confidence on correct predictions to 0.7828, while reducing it on incorrect predictions to 0.6616.

5.2 The Insufficiency of Inference-Only Paradigms

Common NLP downstream tasks can currently be solved by relying on in-context learning coupled with few-shot examples (Brown et al., 2020). This narrative of universal adaptability without requiring data-driven weight updates is frequently applied to user-behavior generation and prediction tasks (Larooij and Törnberg, 2025). However, authentic digital trace data, especially non-English and code-switched text, deviates heavily from original pretraining distributions.

Our results demonstrate a discrepancy between the performance of an in-context, prompt-based approach and data-driven fine-tuning. The empirical finding that a naive statistical heuristic achieves superior calibration and accuracy compared to off-the-shelf models provides evidence against the assumption that generalized inference is universally sufficient for human simulation.

If a simple mathematical baseline is cheaper, more transparent, and achieves a safer confidence distribution than unaligned open-weight models, utilizing off-the-shelf LLMs via standard inference is inadequate for rigorous modeling (Shen et al., 2025a). Therefore, our results suggest that training parameter-efficient models and adapting them specifically for the individual tasks is a methodological requirement.

5.3 Accessibility needed?

Measuring probabilistic alignment, which we deem relevant for evaluation of behavior simulations, requires unrestricted access to a model's internal probability mass. For instance, while the proprietary `Gemini 3.5 Flash-Lite` achieves a competitive discrete accuracy of 0.6868 on our dataset, its

restricted API obscures the underlying logit scores. Consequently, calibration metrics like ECE or Brier scores cannot be computed.

This limitation reduces black-box evaluation to deterministic point-estimates, without associated uncertainty. For rigorous downstream evaluation, accessing the raw computational graph is a methodological prerequisite. Future work must bridge this gap by applying mechanistic interpretability methods (Dunefsky et al., 2024), such as circuit analysis or sparse autoencoders, to open-weight architectures to formally dissect how generative agents construct confidence.

5.4 The Case for Confidence in Generative Agents

We make the case that when benchmarking the ecological and operational validity of generative agents, researchers must explicitly test for confidence alongside accuracy. However, we acknowledge that current methods for extracting confidence in open-text generation are insufficient and underresearched, particularly within the intersection of Computational Social Science and LLMs.

Despite these current methodological limits, shifting from discrete deterministic accuracy to probabilistic evaluation is necessary. Establishing standardized calibration approaches would allow researchers to quantify the robustness of simulations by aggregating individual confidence estimates. Ultimately, recognizing and evaluating the inherent uncertainty in human digital trace data will lead to more sound and transparent computational social science applications.

6 Conclusion

RQ_1: LLMs on the Task Our empirical evaluation demonstrates that off-the-shelf models show lower performance than naive statistical estimations based on the marginal like/dislike distributions, a baseline neither off-the-shelf model beats on accuracy or on calibration. This discrete inaccuracy is accompanied by a confidence signal that does not track it: across both model scale and context length, accuracy and confidence signal move in opposite directions, so higher accuracy alone does not result in better probabilistic alignment.

RQ_2: Impact of Fine-Tuning The impact of task-specific fine-tuning is so significant that data-driven alignment appears necessary for the noisy task at hand. It marks the difference between underperforming the statistical baseline via in-context learning, and outperforming the baseline significantly after training the model on the task. Crucially, it also shows the overarching benefit of achieving closer agreement between measured confidence and observed accuracy.

6.1 Future Work

Expanding to Open Text Our current methodology strictly bounds the action space to a binary classification task to extract and isolate discrete token probabilities. Expanding this framework to generative, open-text simulations requires advancing beyond single-token estimations to accurately capture the behavioral diversity in authentic digital traces.

Advanced Approaches for Modeling Uncertainty To achieve this, future research must evaluate the calibration of generative agents using more complex paradigms for modeling uncertainty. To circumvent the limitations of discrete token probabilities, internal state-based methods present a promising research direction.

Multi-Turn Calibration Currently, our evaluation is constrained to single-turn predictions conditioned exclusively on empirical ground-truth histories. To facilitate complex computational social simulations, models must be capable of multi-turn interactions, autoregressively conditioning on their own synthetic outputs. Future research must evaluate calibration in these dynamic environments, specifically investigating how to quantify and mitigate the compounding uncertainty that occurs during sequential generation.

Testing on Out-of-Distribution (OOD) Data Addressing these structural challenges paves the way for robust, ecologically valid generative agents. Ultimately, the community must benchmark probabilistic alignment on out-of-distribution data, as the generalizability of current models across diverse behavioral patterns remains uncertain. Because a truly operational simulation must reliably generate insights across counterfactual scenarios, achieving robust confidence estimates on OOD data during multi-turn interactions remains the central open challenge in human behavior simulation.

Limitations

Generalizability Across Models and Domains As a case study, our findings are bounded to a single open-weight model family (Ministral 3) evaluated on a single dataset. While this focus enables a controlled analysis of scale and fine-tuning within one architecture, the overconfidence we observe is established for this specific setting rather than for open-weight LLMs in general. Consequently, the calibration deficit and its correction under fine-tuning do not automatically translate to other model families or other forms of noisy behavioral data, which we leave to future work.

Constraint to Discrete Action Spaces A primary limitation of our current methodological framework is its strict reliance on a bounded, binary action space. While scaling this predictive paradigm to multi-class discrete states is mathematically straightforward, quantifying confidence in open-ended text generation introduces complexity. In unconstrained generative environments, foundational metrics such as perplexity serve as inadequate proxies for uncertainty. Consequently, our findings regarding token-level calibration on single-token outputs do not automatically translate to the confidence estimation of free-text behavioral simulacra.

Single-Turn Evaluation Scope Furthermore, our current experimental design is constrained to single-turn inference, evaluating predictions conditioned exclusively on empirical, ground-truth historical interactions. This isolates baseline calibration but leaves multi-turn performance unaddressed. We do not evaluate the model's capacity to maintain probabilistic alignment when autoregressively conditioning on its own synthetic prior outputs. The compounding nature of uncertainty over sequential generations remains a critical unknown; it is not yet established how to calibrate a multi-turn model once its contextual history inevitably diverges from authentic digital trace data.

Ethical Considerations

Behavioral Prediction as a Dual-Use Capability Our task, by design, asks an LLM to infer a specific, identifiable person's future endorsement from a fragment of their prior digital behavior. Framed as a calibration study, this is a methodological probe; framed outside that context, it is a template for individual-level preference profiling. Our central finding, that fine-tuning not only raises accuracy but also sharpens confidence, is what would make such a system attractive for downstream deployment (e.g., comment ranking, engagement optimization, or targeted content curation). A well-calibrated predictor of what a person will approve of is a more trustworthy-looking, and therefore more readily deployable, tool for behavioral targeting than an overconfident one. We see the same calibration intervention we advocate for methodologically as one that lowers the barrier to real-world use of this capability.

Miscalibrated Confidence as a Deployment Risk Beyond privacy, our results affect research and deployment practices: off-the-shelf models are not only inaccurate but also assign high confidence to their own errors (Section 4). Suppose such models were adopted for behavioral simulation or moderation-adjacent decisions without the calibration intervention step we test here, their incorrect but confident outputs risk being trusted as reliable when they are not. Conversely, evaluating LLMs on this kind of task using in-context learning alone, without a transparent statistical baseline or a fine-tuned comparison point, risks the opposite error: concluding that LLMs are broadly unsuited to modeling noisy human behavior when the deficit lies in the inference-only paradigm rather than in the underlying model. Reporting calibration alongside accuracy, and reporting both off-the-shelf and fine-tuned results against a naive baseline as we do, helps guard against both of these premature conclusions.

Positioning within Low-Resource NLP Luxembourgish remains under-resourced for NLP despite growing data availability (Section 2). Contributing an evaluation of confidence estimation on authentic Luxembourgish digital trace data supports this community's research infrastructure and, in principle, downstream applications such as improved comment moderation tooling for the platform that provided the data. We consider this a genuine benefit that should be weighed against the risks above, but not one that offsets them by default: the same methodology, applied to a larger or less consenting population, would carry the profiling risks described above with fewer of the mitigating factors (small-scale, agreement-bound, non-public data) present in our setup.

Acknowledgements

This work was conducted within the Ecological Validation of Artificial Simulacra (EVAS) Research Cooperative (https://evas-research.github.io/). This work is supported by EMO-SCI (project number 569156112) funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation). We thank RTL for providing us with the data for research purposes.

References

Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. 2025. Rethinking Uncertainty Estimation in Natural Language Generation. In *ICLR Workshop: Quantify Uncertainty and Hallucination in Foundation Models: The Next Frontier in Reliable AI*.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand. Association for Computational Linguistics.

Glenn W. Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1 – 3.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, pages 1877–1901, Red Hook, NY, USA. Curran Associates Inc.

Yun-Shiuan Chuang, Agam Goyal, Nikunj Harlalka, Siddharth Suresh, Robert Hawkins, Sijia Yang, Dhavan Shah, Junjie Hu, and Timothy Rogers. 2024. Simulating opinion dynamics with networks of LLM-based agents. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3326–3346, Mexico City, Mexico. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *Preprint*, arXiv:2110.02861.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable LLM feature circuits. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, volume 37 of *NIPS '24*, pages 24375–24410. Curran Associates Inc.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Fernand Fehlen, Peter Gilles, Louis Chauvel, Isabelle Pigeron-Piroth, Yann Ferro, and Etienne Le Bihan. 2023. RP2021 N°8 – Linguistic Diversity on the Rise. Technical report, STATEC and University of Luxembourg, Luxembourg.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Kristina Gligorić, Tijana Zrnic, Cinoo Lee, Emmanuel Candès, and Dan Jurafsky. 2025. Can unconfident LLM annotations be used for confident conclusions? 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)*, pages 3514–3533, Albuquerque, New Mexico. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1321–1330. JMLR.org.

Jake M. Hofman, Amit Sharma, and Duncan J. Watts. 2017. Prediction and explanation in social systems. *Science*, 355(6324):486–488.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Maik Larooij and Petter Törnberg. 2025. Validation is the central challenge for generative social simulation: a critical review of llms in agent-based modeling. *Artificial Intelligence Review*, 59(1):15.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: eliciting truthful answers from a language model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. Ministral 3. *Preprint*, arXiv:2601.08584.

Cedric Lothritz, Bertrand Lebichot, Kevin Allix, Lisa Veiber, Tegawende Bissyande, Jacques Klein, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. 2022. LuxemBERT: Simple and practical data augmentation in language model pre-training for Luxembourgish. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5080–5089, Marseille, France. European Language Resources Association.

Yuxuan Lu, Jing Huang, Yan Han, Bingsheng Yao, Sisong Bei, Yaochen Xie, Yisi Sang, Qi He, and Dakuo Wang. 2026. Can LLM agents simulate multi-turn human behavior? evidence from real online customer behavior data. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 43961–43977, San Diego, California, United States. Association for Computational Linguistics.

Anne-Marie Lutgen, Alistair Plum, Christoph Purschke, and Barbara Plank. 2025. Neural Text Normalization for Luxembourgish Using Real-Life Variation Data. In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 115–127, Abu Dhabi, UAE. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Simon Münker, Achim Rettinger, and Damian Trilling. 2026a. Challenging the myth: A research arc on LLMs as human simulacra. In *Proceedings of The Big Picture v2: Crafting a Research Narrative*, pages 31–44, San Diego, CA, USA. Association for Computational Linguistics.

Simon Münker, Nils Schwager, and Achim Rettinger. 2026b. Don't trust generative agents to mimic communication on social networks unless you benchmarked their empirical realism. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1141–1151, Rabat, Morocco. Association for Computational Linguistics.

Manh Nguyen, Sunil Gupta, and Hung Le. 2026. Probabilities Are All You Need: A Probability-Only Approach to Uncertainty Estimation in Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):32546–32554.

Judith A. Ouellette and Wendy Wood. 1998. Habit and intention in everyday life: The multiple processes by which past behavior predicts future behavior. *Psychological Bulletin*, 124(1):54–74.

Fred Philippy, Siwen Guo, Jacques Klein, and Tegawendé F. Bissyandé. 2024a. LuxEmbedder: A Cross-Lingual Approach to Enhanced Luxembourgish Sentence Embeddings. *arXiv preprint*. ArXiv:2412.03331 [cs].

Fred Philippy, Shohreh Haddadan, and Siwen Guo. 2024b. Forget NLI, Use a Dictionary: Zero-Shot Topic Classification for Low-Resource Languages with Application to Luxembourgish. In *Proceedings of the 3rd Annual Meeting of the Special Interest Group on Under-resourced Languages @ LREC-COLING 2024*, pages 97–104, Torino, Italia. ELRA and ICCL.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alistair Plum, Laura Maria Bernardy, and Tharindu Ranasinghe. 2026. Do llms judge distantly supervised named entity labels well? constructing the judgewel dataset. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, pages 7707–7718, Palma, Mallorca, Spain. European Language Resources Association (ELRA).

Alistair Plum, Tharindu Ranasinghe, and Christoph Purschke. 2025. Text Generation Models for Luxembourgish with Limited Data: A Balanced Multilingual Strategy. In *Proceedings of the 12th Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 93–104, Abu Dhabi, UAE. Association for Computational Linguistics.

Christoph Purschke. 2020. Attitudes Toward Multilingualism in Luxembourg. A Comparative Analysis of Online News Comments and Crowdsourced Questionnaire Data. *Frontiers in Artificial Intelligence*, 3:536086.

Christoph Purschke. 2025. Discourse figures in the Luxembourg language debate (2015–2020). *Zeitschrift für Sprachvariation und Soziolinguistik*, 1(2).

Tharindu Ranasinghe, Alistair Plum, Christoph Purschke, and Marcos Zampieri. 2023. Publish or hold? Automatic comment moderation in Luxembourgish news articles. In *Proceedings of the 14th international conference on recent advances in natural language processing*, pages 968–978, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Nils Schwager, Simon Münker, Alistair Plum, and Achim Rettinger. 2026. Towards simulating social media users with LLMs: Evaluating the operational validity of conditioned comment prediction. In *The Proceedings for the 15th Workshop on Computational Approaches to Subjectivity, Sentiment Social Media Analysis (WASSA 2026)*, pages 208–221, Rabat, Morocco. Association for Computational Linguistics.

Hua Shen, Tiffany Knearem, Reshmi Ghosh, Michael Xieyang Liu, Andrés Monroy-Hernández, Tongshuang Wu, Diyi Yang, Yun Huang, Tanushree Mitra, Yang Li, and Marti Hearst. 2025a. Bidirectional Human-AI Alignment: Emerging Challenges and Opportunities. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '25, pages 1–6, New York, NY, USA. Association for Computing Machinery.

Hua Shen, Tiffany Knearem, Reshmi Ghosh, Yu-Ju Yang, Nicholas Clark, Tanu Mitra, and Yun Huang. 2025b. ValueCompass: A framework for measuring contextual value alignment between human and LLMs. In *Proceedings of the 9th Widening NLP Workshop*, pages 75–86, Suzhou, China. Association for Computational Linguistics.

Vaishnavi Shrivastava, Percy Liang, and Ananya Kumar. 2023. Llamas know what gpts don't show: Surrogate models for confidence estimation. *Preprint*, arXiv:2311.08877.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore. Association for Computational Linguistics.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Tianyu Wang, Jiajun Li, and Jianghao Lin. 2026. Can llms think like consumers? benchmarking crowd-level reaction reconstruction with consumersimbench. *Preprint*, arXiv:2605.17079.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024. "my answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7407–7416, Bangkok, Thailand. Association for Computational Linguistics.

Wanyi Zhang, Qiang Shen, Stefano Teso, Bruno Lepri, Andrea Passerini, Ivano Bison, and Fausto Giunchiglia. 2021. Putting human behavior predictability in context. *EPJ Data Science*, 10(1):42.

A Code and Model Availability

The technical pipeline and source code are publicly available in the following repository: `https://github.com/nsschw/accuracy-is-not-enough`. To mitigate potential misuse while ensuring reproducibility, fine-tuned models are restricted to scientific use and shared only upon request. This policy aligns our open science commitment with responsible research practices.

From Errors to Precision: A Taxonomy-Guided Approach for LLM Flood Cause Extraction

Jane Arleth dela Cruz **Iris Hendrickx** **Martha Larson**
Center for Language and Speech Technology
Center for Language Studies
Radboud University, Nijmegen, Netherlands
{janearleth.delacruz, iris.hendrickx, martha.larson}@ru.nl

Abstract

Large language models (LLMs) offer disaster risk management (DRM) practitioners a promising "plug-and-play" solution for information extraction on online media data. However, they can struggle with the strict precision required for identifying semantically and structurally complex disaster causes. We evaluate how off-the shelf LLMs extract cause information from flood-related texts and found a strong recall bias. To diagnose these errors, our primary contribution is a comprehensive, human-annotated qualitative error taxonomy using inductive and deductive coding. We identified specific linguistic traps, i.e., impact descriptions, background context, and subjective opinions, that trigger over-extraction, indicating models have a patterned failure. As an exploratory second step, we apply this taxonomy to design an ablation study of prompt-based interventions using definitions and exclusions. Our results show that our approach successfully shifts models along the precision–recall trade-off, i.e., curbing over-extraction, improving precision and domain-specific F0.5-score for recall-biased models. We demonstrate that by leveraging qualitative error analysis, domain experts can not only characterize and refine off-the-shelf LLMs, but also navigate the tradeoffs of prompting.

1 Introduction

The emergence of off-the-shelf large language models (LLMs) offer disaster risk management (DRM) domain experts the potential independence from complex NLP pipelines and the maintenance they imply (McDaniel et al., 2024; Lei et al., 2025; Imran et al., 2025). The "plug-and-play" nature of these LLMs allow domain experts to rapidly deploy customizable information extraction pipelines without the need for fine-tuning or annotating domain-specific large training datasets.

While recent surveys highlight the broad potential of LLMs in DRM (Xu et al., 2025; Lei et al., 2025), studies are heavily focused on situational awareness (Fu et al., 2022; Chowdhury et al., 2024; Peng and Zhu, 2025), impact classification (Alam et al., 2021; Imran et al., 2025; McDaniel et al., 2024; Qian et al., 2024), and rescue prioritization (Koju et al., 2024). Cause extraction, the task of identifying the underlying reasons why a disaster event occurred, remains under-explored (Duong et al., 2025; Peng and Zhu, 2025), despite it being essential for post-event accountability, risk analysis, and long-term mitigation planning. By integrating causal statements from multiple platforms such as noisy user-generated social media posts and online news reporting, practitioners and decision-makers can complement official reports with online observations to build a more complete post-event report. Cause extraction is a fundamentally difficult NLP challenge due to the semantic and structural complexity of causal language (Solstad and Bott, 2017; Garcia-Corral et al., 2025; Kiciman et al., 2024; Jin et al., 2024).

In this paper, we study flood cause extraction: identifying why a flooding event occurred from event-related online documents. We operationalize this task at two granularities (Figure 1) (1) document-level flood cause classification (document classification), predicting whether a document contains a flood cause, and (2) sentence-level flood cause extraction (sentence extraction), identifying which sentence(s) contain that cause in positive documents.

Our initial evaluations reveal that the LLMs are recall-biased: they often classify flood-related but non-causal information as causal. In flood reporting, causes are often co-located with impact information, background context, speculation, and causal direction is easily a source of confusion: statements where the flood is the cause of an impact are very common, (e.g., `flash floods inundate homes`, `flooding caused evacuation`) and the underlying causes (e.g., `blocked drainage`, `dam`

Proceedings of the 11th Workshop on Natural User-generated Text, pages 101–116
October 28, 2026 ©2026 Association for Computational Linguistics

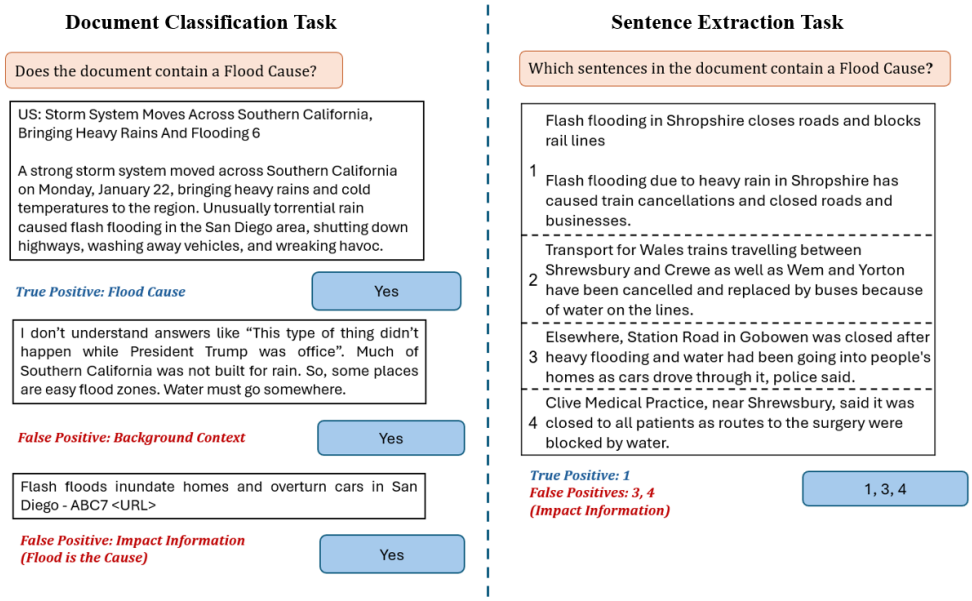


Figure 1: We investigate LLMs' performance in two tasks (1) **Document-level flood cause classification (Document Classification - Left)**, which asks whether the document contains information about the cause of the flood, and (2) **Sentence-level flood cause extraction (Sentence Extraction - Right)**, which asks the model to identify the specific sentences in the document that contain the causal information.

`failure`) can be buried. In crisis informatics, while extracting *impacts* naturally favors high recall to ensure no calls for rescue or response are missed, extracting *causes* requires strict precision. A low precision system would mean increased review effort needed for practitioners for filtering false positives. Although missed cause statements remain undesirable, our use case is characterized by the inherent event-level redundancy across documents.

To systematically address these limitations, we argue that simple prompt engineering must be grounded in rigorous, qualitative error analysis. Aggregate quantitative metrics fail to capture specific semantic and factual errors of generative models, demonstrated by Lux et al. (2020) in the context of abstractive summarization. Inspired by this approach, our primary contribution is a comprehensive, human-annotated error taxonomy of flood cause extraction failures in a multi-platform flood dataset. We carried out qualitative coding on the baseline false positives of LLMs in both the document and sentence-level granularities, where we identified overarching triggers for over-extraction. We create an error taxonomy using both inductive and deductive coding of the error categories. Beyond identifying model error categories, the taxonomy also reveals where error interpretation becomes difficult for humans, particularly in sentence-level judgements.

As a secondary and exploratory contribution, we use this error taxonomy as a blueprint to design a study of customized prompt interventions. We evaluate whether providing LLMs two types of prompt constraints: (1) explicit definitions (telling the model what to look for) and (2) exclusion rules (telling the model what not to look for) can reliably customize the LLMs' baseline zero-shot performance.

Our initial prompt-based experiments demonstrate that our taxonomy-driven constraints successfully shift models from greedy over-extraction to a more selective extraction, i.e., improving precision and the domain-relevant F0.5-score. Explicit exclusion rules consistently outperform definition guidance, and the strongest gains are concentrated in the Forecast and Warning and Background Context categories. We suggest that reliable LLM use on noisy user-generated text requires moving beyond aggregate scores: a domain-specific qualitative error analysis is necessary to characterize recurring failures to establish what model errors are, where the limitations of LLMs lie, and how interventions can be designed and assessed.

2 Related Work

Cause Extraction. Extracting causes is not just a DRM gap; rather, it is a fundamental challenge

in the broader NLP domain due to the semantic and structural complexity of causal language (Solstad and Bott, 2017; Garcia-Corral et al., 2025). Most work is focused on scientific literature, news, and finance (Garcia-Corral et al., 2025). Recently, LLMs have been evaluated for their causal and reasoning capabilities. Jin et al. (2024) demonstrated LLMs' failure to generalize causal inference in out-of-distribution settings and Kiciman et al. (2024) found their reasoning capabilities break down in ambiguous contexts. Our work is closely related to Garcia-Corral et al. (2025) who investigated causal extraction in the subjective domain of political context, through causal sequence classification and causal span detection, and revealed that LLMs can be "fooled" by specific linguistic features.

Qualitative Error Taxonomy Creation. As generative modes scale in complexity, automated quantitative metrics are no longer sufficient for evaluation. Standard metrics (e.g., BLEU, ROUGE) can mask the LLMs' nuanced, semantic errors (Lux et al., 2020; Xu et al., 2022; Nainia et al., 2025). Other existing automatic evaluation metrics does provide partial answers: such as embedding-based metrics like BERTScore (Zhang et al., 2020) and BARTScore (Yuan et al., 2021) that estimate semantic similarity. Although useful indicators for general quality, these metrics cannot directly assess the type of errors. To understand why LLMs fail beyond these metrics, human-driven failure analysis and error taxonomy creation work has been done in the software engineering field (Wang et al., 2025; Nekrasov et al., 2026; Rahman et al., 2020). Wang et al. (2025) found that modern LLMs have adequately learned the syntax rules but struggle with the semantic requirements of programming. Although there is a trend of using LLMs for self-evaluation and error analysis (Singh et al., 2026), this can introduce new biases and reinforce the need for human oversight (Otero et al., 2025).

Prompt Interventions. Practitioners commonly adapt LLMs to domain-specific tasks using general-purpose prompting strategies, including reasoning-oriented methods, most notably, Chain-of-Thought (CoT) prompting (Kojima et al., 2022; Wei et al., 2022; Yao et al., 2023), have become standard methods for improving adaptability in complex NLP tasks. In disaster management classification tasks, rationale-generation during prediction has been proposed (dela Cruz et al., 2025), because it offers textual justifications that allows human oversight while few-shot prompting has been an important intervention for improvement by allowing users to implicitly communicate task requirements through labeled examples (Brown et al., 2020; Taghian Dinani et al., 2024; Imran et al., 2025). We compare rationale-generation with prediction (Rat) and few-shot prompting (FS) as general-purpose strategies against our taxonomy-guided prompts.

Prompt constraints offer a model improvement strategy that shapes model behavior through direct instructions rather than implicit examples and broad reasoning steps (Chen et al., 2024; Lu et al., 2023). Automatic prompt optimization can also improve performance (Singh et al., 2026; Yang et al., 2024) but can not only introduce new biases but also reduce user agency needed for high-stakes tasks. Following Imran et al.'s (2025) insight on the need for qualitative analysis for understanding why LLMs can fail in crisis contexts, we translate our qualitative error taxonomy into targeted prompt constraints.

3 Experimental Setup

Dataset. We conduct our experiments on a private flood cause dataset developed in collaboration with a company that operates social-media crisis monitoring and automated reporting pipelines. The dataset comprises of a curated set of documents from four online platforms: News, X (formerly Twitter), YouTube, and Bluesky, reporting on 20 flood events that occurred between 2022-2025 collected from Dela Cruz et al.'s (2026) study. The dataset provides a highly realistic event-centered candidate pool that accurately reflects the noisy data encountered in true operational workflows. Each flooding event is defined by its location and time period and contains between 20 and 100 documents. Although the dataset cannot be released due to company ownership, our investigation setting reflects a realistic, noisy, multi-platform context used in practice.

The dataset was manually annotated at two levels of granularity. A document receives a positive label if it contains objective information about a flood's cause or contributing factors; labels exclude mentions of flood impacts, aftermath, flood types, and general weather forecast as these are not the causes of the flood. For positive documents, the specific causal sentence(s) are also identified. These labels support the two tasks: (1) document classification and (2) sentence extraction. We partition the data into two event-level splits (see Table 1): the de-

velopment set containing the 10 events for initial benchmarking and error taxonomy construction, and the validation set of 10 events for validation of prompt interventions.

Data Split	Flood Events	Docs	Docs w/ Cause	Sent	Sent w/ Cause
development	10	600	305	2,912	452
validation	10	835	294	2,133	405

Table 1: Data splits for error taxonomy creation (development set) and prompt intervention experiments (validation set).

Models. We evaluate four commonly used LLMs: GPT 5-mini (OpenAI, 2025), GPT 4o-mini (OpenAI, 2024), Llama 3.1-8B-instruct (Llama Team, 2024), and Mistral Nemo (Mistral AI Team, 2024). These LLMs were selected to include both closed and open-weight systems, enabling comparison across model families. More details are found in Appendix A.

4 Error Taxonomy Creation

In this section, we carry out inductive coding at the document-level of the false positives in the development set to construct an **error taxonomy**. Then, we use deductive coding to validate whether the taxonomy is applicable at the sentence-level.

4.1 Inductive Coding at the Document-level for Taxonomy Creation

Table 2 shows the baseline performance of the LLMs in the document-level flood cause classification task. We observe that the models are mostly recall-biased, having as high as 41 points precision-recall gap (PR-Gap) from GPT 5-mini, indicating that the models often over-predict flood causes.

We carried out inductive coding on the false positives in our development set. We sampled a total of 176 documents, prioritizing the cases where at least two models were incorrect (140 documents) and adding model-specific failures at 10 documents per model except for Mistral (6 documents). Inductive coding is a qualitative research approach of developing codes "directly" from the data by developing codes using phrases or terms in data, rather than using the theoretical or vocabulary of the researcher (Skjott Linneberg and Korsgaard, 2019). We looked for generalizations, patterns, outliers, and salient themes in the sample of false positives, by initially assigning open descriptive codes that answer the question: *what information does the document contain?* Through iterative grouping of the initial codes, we consolidated the errors into five overarching categories. Figure 2 shows the distribution of the errors in the sample.

Flood Cause Document Classification					
Model	A	P	R	PR - Gap ↓	F0.5-score
GPT 4o-mini	77	67	97	30	71
GPT 5-mini	69	59	100	41	64
Llama 3.1	70	61	93	32	66
Mistral Nemo	72	68	69	1	68
Flood Cause Sentence Extraction					
Model	A	P	R	PR - Gap ↓	F0.5-score
GPT 4o-mini	83	73	91	18	76
GPT 5-mini	83	71	96	25	75
Llama 3.1	71	59	92	33	64
Mistral Nemo	79	70	79	9	72

Table 2: Baseline performance of LLMs in both the flood cause document classification and sentence extraction tasks on the development set. Performance is measured on Accuracy (A), Precision (P), Recall (R), F0.5-score, and the Precision-Recall Gap (PR-Gap).

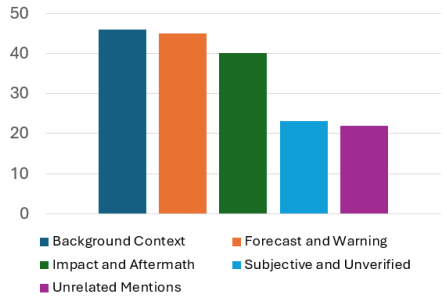


Figure 2: Distribution of false positives in the Document-level error sample.

We hypothesize that LLMs misinterpret **Impact and Aftermath** information, such as "`Residents 'evacuated' from rural homes near Chester due to flooding`", by treating impact descriptions (outcomes), like damage or evacuation as causes perhaps due to the presence of causal relation but invert the cause-and-effect relationship. Similarly, LLMs mistake general **Background Context** for specific flood event triggers, overgeneralizing location descriptions like, "`Hail storm and a record amount of precipitation and lightning in Valencia city`" even though no flooding event is described. They also stumble on weather **Forecast and Warnings**, where they falsely interpret predictive language like "`Heavy rainfall today may cause some flash flooding across Scotland`" as factual report-

ing post-event.

In addition to contextual and directional errors, LLMs rely on explicit lexical cues and have broad semantic scope. They accept **Subjective and Unverified** causality, e.g., "`Well, at least California has destroyed flood control dams so more natural water flow fills bays so some freaking snail survives`" as valid because they fail to differentiate objective reporting from speculation, opinion and sarcasm. Lastly, LLMs can be fooled by superficial keyword matches we call **Unrelated Mentions** and fail to differentiate natural hazard context use of flood from metaphorical or non-hydrological use of the word flood. For example, an LLM fails in the text: "`Eight people die in the London Beer Flood, an accident at Meux & Co's Horse Shoe Brewery`" to restrict the semantic scope to hydrological events.

After the inductive coding established the error taxonomy, we assessed the reliability. A second annotator independently labeled a 20% sample of the false positives in our development set (42 documents) using the error taxonomy. The Cohen's kappa agreement was $\kappa = 0.68$ for the document-level error category classification, indicating substantial agreement according to the Landis and Koch's (1977) scale.

4.2 Deductive Coding of Sentence-Level Errors for Validation

We investigated whether our taxonomy is applicable at the sentence-level extraction task. We used a deductive coding approach to annotate the extracted false positive sentences onto the five categories established in the inductive process. Deductive coding uses pre-defined list of codes created before coding the data (Skjott Linneberg and Korsgaard, 2019). Since sentence extraction is only performed on documents already classified as containing causal information, we sampled a new set of 100 unique sentences where all four LLMs agreed on an incorrect extraction. By coding this new sample, we tested for qualitative saturation. Our error taxonomy successfully captured most of the errors, and thus demonstrates its robustness.

During this process, we identified a new error category unique to the sentence-level task, which we call *weak causality*. These are sentences that contain possible physical cause of the flood but lack the context and connection to the flood when viewed in isolation. The models extract vague pronouns or generic sentences of weather statements (e.g., `They are caused by Hurricane...`, `This occurred after prolonged heavy rainfall...`).

Annotation agreement was at Cohen's $\kappa = 0.62$ for the sentence-level error category classification. It was lower than the document-level with most of the disagreements between *background context* and *weak causality* categories.

5 Prompt Interventions

In this section, we test whether the error taxonomy can be used to control model behavior by translating the error categories into **taxonomy-guided prompt interventions**. Our prompt design was specifically for document classification that the error taxonomy was developed from.

5.1 Prompts

Table 3 describes the groups of prompts used for the flood cause document classification: (1) our taxonomy-guided prompt interventions and (DEF, EX, COMB, Table 3 rows 2-4) (2) general-purpose prompt interventions that practitioners use that are not based on a qualitative failure analysis (RAT, FS, Table 3 rows 4,5). We followed our error taxonomy to design taxonomy-guided prompt interventions. We mapped our specific error categories to two types of instruction: explicit definitions, telling the model what to look for and exclusion rules, telling the model what *not* to look for. We present the types of instructions that would best resolve each error category:

- Explicit Definition of Flood Cause - Errors from the categories: Background Context and Unrelated Mentions would require clear operationalization of causality. By defining what a flood cause as *why* the flooding happened and the *contributing factors* to the flood, the model can be guided to ignore background context statements and unrelated discussions.
- Exclusion Rules - Errors from the Categories: Impact information and Aftermath, and weather Forecast and Warnings would require negative constraints. We explicitly instruct the models to exclude Impacts and Forecast despite possible presence of causal syntax (e.g. `flooding caused damage to houses`)
- Combined Approach - Errors concerning Subjective and Unverified Causality are complex

Prompts	Description
Baseline (BASE)	The prompt is taxonomy-naive. It asks whether the document the text contains a flood cause or not.
Definition-aware (DEF)	The prompt augments Base with an explicit natural language definition of what constitutes as a flood cause or contributing factors to the flood. It is designed to suppress background context and unrelated mentions categories, where the primary failure is the absence of a clear causal boundary.
Exclusion-aware (EX)	The prompt augments Base by providing explicit negative constraints for all error categories except for unrelated mentions. impact and aftermath, Forecast and warnings, and subjective and unverified are targeted because of causal surface syntax.
Combined (COMB)	The prompt combines both Def and EX instructions to Base simultaneously. It represents the full application of the error taxonomy.
Rationale-generation with prediction (RAT)	The prompt instructs the model to produce an extractive rationale explaining why a document qualifies as a flood cause before committing to the decision. We used the explain-then-predict (E →P) version, adopted from dela Cruz et al.'s (2026) approach on integrating rationales in the classification task for the disaster management domain.
Few-shot (FS)	The prompt provides the model with a small set of labeled example documents (5 documents) drawn from the development set. Negative examples are non-causal documents sampled to cover four of the five taxonomy categories (no unrelated mentions).

Table 3: DEF, EX, AND COMB prompts make the error taxonomy explicitly visible to the model through definition or exclusion rules. RA and FS prompts are general-purpose prompt interventions used for model improvement. They represent intervention strategies a practitioner would reach for without qualitative failure analysis.

as they require both clear definition of an objective cause and somewhat explicit exclusion opinions and speculations.

The full prompts we used incorporating the types of instructions can be found in Appendix B.

5.2 Evaluation and Scope

We evaluate the aggregate impact of our interventions for both document classification and sentence extraction but restrict our deeper behavioral analysis in the document-level. For each taxonomy-guided prompt condition $p \in \{\text{DEF, EXC, COMB}\}$ (see Table 3, rows 2-4), we compute the precision, recall, F0.5-score (which is precision-biased), and the precision-recall gap on the validation set on both the document classification and sentence extraction. Our deep behavioral analysis, specifically the per-error category suppression analysis and comparison against general-purpose interventions for classification (RAT and FS, (see Table 3, rows 4,5)), is restricted to the document-level. Because sentence-level extraction introduces granularity-specific challenges such as *weak causality*, we prioritize solving the foundational document-level classification task first and leave extraction-specific behavioral analysis to future work.

We annotate the false positives produced by each baseline model (BASE) on the validation set via deductive coding using the error taxonomy established in Section 4. Each false positive is assigned exactly one label; in cases where a document or sentence contains more than one category, we assigned the most dominant error type. For each taxonomy category, we calculate the proportion of baseline false positives reduced (False Positive Reduction Rate (FPRR) by each prompt condition.

Let $\text{FP}_c^{(p)}$ denote the number of false positives assigned to category c under the prompt condition p. For each category $c \in C$, where C is the set of error taxonomy categories, we compute:

Absolute False Positive Reduction:

$$\Delta_c^{(p)} = \text{FP}_c^{(\text{Base})} - \text{FP}_c^{(p)}$$

False Positive Reduction Rate:

$$\text{FPRR}_c^{(p)} = \frac{\Delta_c^{(p)}}{\text{FP}_c^{(Base)}}$$

where $\text{FPRR}_c^{(p)} \in [0, 1]$ with 1 indicating complete reduction of category c under condition p. We consider an intervention to behave as intended if the highest values of $\text{FPRR}_c^{(p)}$ are concentrated in the categories of its design targets.

Since our per-category error counts are small, this analysis is treated as **indicative**.

5.3 Results

Our experiments demonstrate that taxonomy-guided prompt interventions can shift model behavior towards higher precision and outperform existing general-purpose approaches for the high-precision flood cause document classification.

Taxonomy-guided prompt interventions can shift the precision-recall trade off. We present the aggregate results of the prompt interventions in the validation set in Table 4. At the document-level, we observe the models shift along the precision-recall trade off characterized by the decrease in the Precision-Recall gap (PR-Gap), more consistently for Exclusion (EX) and Combined (COMB)

prompts across models. The improvements in F0.5-score were driven mainly by the EX and COMB approaches, while DEF yields little to no improvement.

Sentence-level extraction results show the same general pattern (see Table 9, where EX and COMB drive the PR-Gap decrease and F0.5-score improvements for all models. Llama 3.1 improved by 8 points in the F0.5-score in the COMB prompt.

Flood Cause Document Classification Task

Model	Prompt	P	R	PR-Gap ↓	F0.5-score
GPT 4o-mini	BASE	62	97	35	67
	DEF	72	90	18	75
	EX	74	83	9	76
	COMB	**79**	77	**-2**	**79**
	RAT	45	**100**	55	50
	FS	57	98	41	62
GPT 5-mini	BASE	57	**100**	43	62
	DEF	56	**100**	44	62
	EX	**67**	97	**30**	**72**
	COMB	63	98	35	68
	RAT	58	**100**	42	63
	FS	60	**100**	40	66
Llama 3.1	BASE	57	92	35	62
	DEF	57	91	34	62
	EX	**65**	77	12	**67**
	COMB	65	76	**11**	**67**
	RAT	48	95	47	53
	FS	48	**98**	50	53
Mistral Nemo	BASE	**71**	59	-12	68
	DEF	69	72	**3**	**69**
	EX	68	62	-6	67
	COMB	72	53	-19	67
	RAT	41	**100**	59	46
	FS	60	80	20	63

Table 4: Performance of LLMs in the document classification task using our prompt constraints on the validation set. Performance is measured on Precision (P), Recall (R), F0.5-score (precision-biased), and the Precision-Recall Gap (PR-Gap). **Bold** values indicate performing prompt for the metric.

Taxonomy-guided prompt interventions selectively suppress targeted errors. Figure 3 shows the category-specific False Positive Reduction Rate (FPRR) for each prompt intervention. Across models, the largest error reductions occur for Forecast and Warnings and Background Context. Def has the weakest effect overall: it produces moderate reductions for GPT-4o-mini, particularly for Background Context and Unrelated Mentions, but has near-zero effect for GPT5-mini and Llama 3.1-8B and produces new false positives for Mistral Nemo.

EX produces the most targeted suppression, with the largest reductions consistently occurring for Forecast and Warnings and Background Context across models. Unrelated Mentions, which EX does not explicitly target, shows smaller reductions. COMB is more model-dependent: it suppresses over 70% of Background Context and Forecast and Warnings errors for GPT 4o-mini, while its profile for GPT 5-mini and Llama 3.1-8B largely follows EX.This suggests that the exclusion component drives most of the behavioral effect.

Taxonomy-guided prompting is only effective when models have the error profile the prompts are designed to address as shown by Mistral Nemo, the outlier in our experiments. Mistral was not recall-biased to begin with. Only Forecast and Warning and Background Context errors are suppressed by EX and COMB prompts.

General Purpose Interventions fail in the high-precision requirements of the task. Table 4 also compares our taxonomy-guided prompts (DEF, EX, COMB) against general-purpose alternatives, rationale-generation (RAT) and few-shot (FS). Our results indicate that general-purpose prompt interventions are not as effective as the taxonomy-guided prompts in improving precision and instead maximize recall.

RAT produced the lowest precision scores across all models and widened the PR-gap compared to BASE, as it had the highest recall in three out of the four models. This indicates that RAT encourages recall of the flood causes.

FS performed better than RAT, but was only better than BASE for GPT 5-mini in terms of F0.5-score. It had the second-highest recall scores next to RAT and generally widened the PR-Gap. The implicit boundaries from the explicit taxonomy-based negative examples were insufficient to improve precision and shift the recall bias.

6 Discussion

We synthesize our findings on why baseline LLMs fail at flood cause extraction, how the error taxonomy can make failures diagnosable and actionable, and why general-purpose prompting strategies struggle in our high precision task.

LLM failures are patterned and diagnosable. Our findings show that the main challenge in causal information extraction is not a simple lack of model capability, rather a fundamental mismatch between the general-purpose design of LLMs and the precise, high-stakes requirements of the DRM task. We reveal that baseline LLMs fail in patterned, linguistically meaningful ways, consistent with pre-

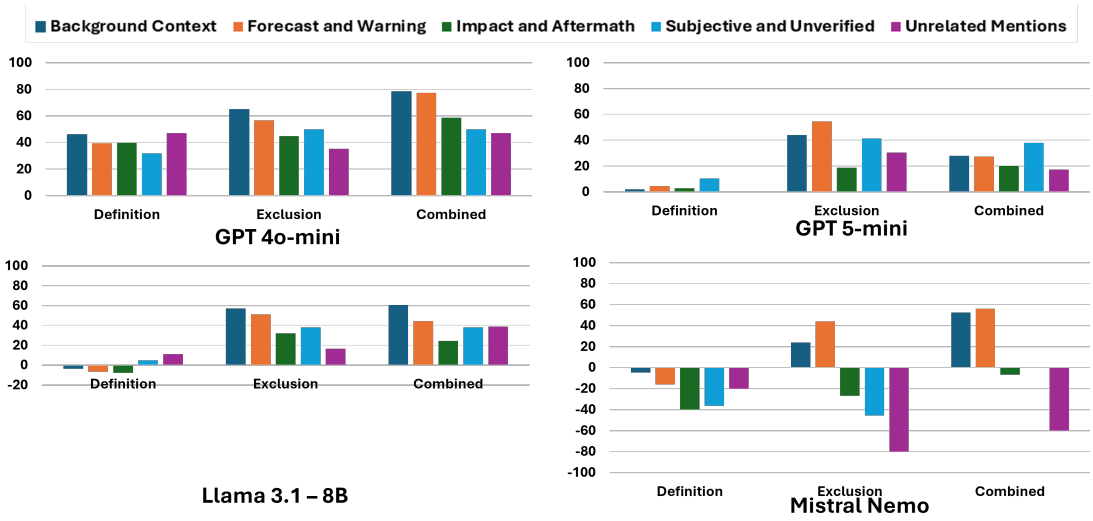


Figure 3: False Positive Reduction Rate for each prompt intervention in flood cause document classification. The values represent the percentage of Baseline false positives eliminated under each new prompt condition. A value of 60% means 60% of baseline false positives were suppressed in that category. Negative values indicate that the intervention introduced more false positives than the baseline.

vious findings (Wang et al., 2025; Garcia-Corral et al., 2025).

Our qualitative error analysis strengthens the point of how low precision is a problem in extraction of flood causes. A recall-biased system can be operationally untrustworthy in our DRM context. Disaster managers would need to review and filter large numbers of incorrect output weakening the value of a supposed "plug-and-play" pipeline.

The core contribution of our study is the error taxonomy. It transforms aggregate model errors into interpretable and actionable categories. By identifying the recurring error categories, we provide domain experts with a structured vocabulary for diagnosing and steering model behavior without requiring technical AI expertise, a way of human-AI collaboration grounded in domain knowledge.

The taxonomy also reveals where human interpretation of LLM errors becomes difficult. While inter-annotator agreement was substantial at the document level $\kappa = 0.68$, it dropped to $\kappa = 0.62$ at the sentence level, with most disagreements occurring between *background context* and *weak causality*. Crucially, this drop in inter-rater agreement at the sentence level reflects disagreements in diagnosing *why* the LLM failed, not whether it failed. This shows a challenge in AI governance: diagnosing LLM failures in real-world narratives, where disagreements are highly context-dependent. The emergence of *weak causality* in the sentence-level analysis reflects that causal meaning is often distributed across discourse and can be lost when the unit of analysis is reduced to a single sentence. Therefore, LLM extraction failures must not be reviewed in isolation.

Taxonomy-guided prompting works, but selectively. The taxonomy-guided prompt interventions improve precision-oriented performance but their effectiveness is not as straightforward and uniform. EX and COMB prompts consistently outperform DEF and improvements are concentrated in more specific error categories i.e., Forecast and Warning and Background Context. Therefore, targeted interventions make the models respond to specific linguistic signals associated with particular error types.

The taxonomy-guided interventions for Mistral Nemo do not help all categories but can harm, which confirms that these strategies are contingent to the baseline error profile of the models. This model dependence means that taxonomy-guided prompting is not a universal application. Practitioners must therefore always first diagnose baseline model performance.

General-purpose interventions fail on the high-precision requirements of the task. RAT and FS prompting failed to match the taxonomy-guided conditions and generally harmed precision. RAT prompt was drawn from prior disaster management domain work using similar social media data, and was therefore the most ecologically valid and interpretable baseline. Therefore, we expected it to

perform competitively but instead it lowers precision and maximizes recall. This reveals a fundamental goal mismatch. The original RAT prompt was designed for impact classification and asking for extractive rationales (snippet that justifies the classification), where broader judgments are useful. Flood cause classification requires a narrower boundary. Without explicit definitions and exclusions, the model may apply a broad prior that relevant flood content justifies a positive label, while the extracted rationale reinforces that decision.

The FS prompt was marginally better than the RAT prompt. We consider the FS prompt as taxonomy-informed, in terms of providing examples per error category. Our results revealed that this implicit boundary prompting strategy through labeled examples is insufficient and ineffective. Together with the findings from the EX prompt, we recommend that the taxonomy must be revealed as a direct instruction and not as demonstrations alone.

7 Conclusion

This paper presents a taxonomy-driven framework for analyzing and mitigating LLM over-extraction of flood cause information. We demonstrate that off-the-shelf LLMs fail in patterned ways, treating impact descriptions, background context, and subjective opinions as causal evidence. By translating the error taxonomy into prompt constraints, we showed taxonomy-guided prompting can steer LLMs toward the high-precision requirements.

Our results carry a broader implication for the deployment of LLMs in any resource-constrained, high-precision domain. The appeal of off-the-shelf LLMs is their accessibility; no labeled training data, no fine-tuning, no specialized AI infrastructure are needed. However, our study shows that this accessibility comes at a cost, as revealed by our rigorous domain-specific error analysis; these systems reflect the general statistical patterns of their training data rather than the precise boundaries required in domain-specific fields. This is especially relevant in disaster management, where recent surveys highlight the promise of LLM-based workflows but also the need for careful adaptation to specific crisis tasks (Lei et al., 2025; Xu et al., 2025).

We argue that qualitative error analysis not only supplements quantitative evaluation, but is also a prerequisite for responsible deployment. Our pipeline of inductive taxonomy construction, deductive saturation testing, and taxonomy-to-prompt translation offers a replicable framework for DRM practitioners and AI developers working across crisis domains.

Limitations

This paper has several limitations. Our dataset is an English-only private dataset collected from News, X (formerly Twitter), YouTube, and Bluesky collected from Dela Cruz et al. (2026). The dataset for cause classification consists of documents that were collected around specific flood events as defined by a proprietary algorithm. We only evaluated off-the-shelf LLMs. Finetuning these models may lead to more favorable results.

The scope of our use case is limited, with only flood cause document classification task and sentence-level extraction task. Our scope was mainly focused on minimizing false positives, although missed cause statements are still undesirable, our use case is characterized by the inherent event-level redundancy across documents, whereas false positives can directly increase practitioner workload and contaminate post-event synthesis.

Our taxonomy creation was based on a single expert annotator and validated with a 20% sample based on another expert annotator. Our human annotator (first author) is fluent in English and has relevant experience in crisis informatics and corpus linguistics. The second annotator (second author) is fluent in English and has relevant experience in corpus linguistics. While the inter-rater reliability scores support the reliability of the annotation scheme, they also suggest that sentence-level cause extraction remains lower than the document-level. The annotation scope was very focused due to the **sensitive nature** of the dataset. Documents were collected from real flooding events and thus contains distressing crisis content of impacts of flooding event such as reports of deaths, injuries, missing persons, rescue needs, infrastructure damage, etc.

Ethical Considerations

Although we cannot release the data, the description of the data construction pipeline and the annotation protocol to facilitate replication on comparable data is found in (Dela Cruz et al., 2026). Our work aspires to support disaster management practitioners. Our work demonstrates measurable

progress in constraining LLM over-extraction behavior of flood causes through taxonomy-driven prompting. However, we observe that even in the best performing prompt configuration, they do not achieve the precision requirements needed for unsupervised deployment pipelines. Therefore, we caution against misinterpreting the readiness of the technology for use, and move either too quickly to uptake without guarantees of reliability.

Conversely, we also caution against premature abandonment of the type of solutions we study. The LLMs we studied exhibit structured and diagnosable failure patterns. This means they are amenable to iterative improvement through methods like those presented here, and may already add value in human-in-the-loop settings where practitioners retain full interpretive control. We strongly recommend supervised integration with domain experts.

We recognize that the annotation work for our task carries emotional burden. The source documents and sentences contained possible highly distressing information regarding the context and impacts of real-world flooding events. Reading and annotating these content carries inherent psychological risks. Annotators were properly informed and warned beforehand, about the sensitive nature of the task prior to participation.

Acknowledgments

This publication is part of the project 'Indeep: Interpreting Deep Learning Models for Text and Sound' with project number NWA.1292.19.399, which is partly financed by the Dutch Research Council (NWO). We would like to thank FloodTags for their collaboration, specifically in providing the case study and access to their dataset.

References

Firoj Alam, Umair Qazi, Muhammad Imran, and Ferda Ofli. 2021. Humaid: Human-annotated disaster incidents data from twitter with deep learning benchmarks. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):933–942.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Yihan Chen, Benfeng Xu, Quan Wang, Yi Liu, and Zhendong Mao. 2024. Benchmarking large language models on controllable generation under diversified instructions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17808–17816.

Md Towhidul Absar Chowdhury, Soumyajit Datta, Naveen Sharma, and Ashiqur R. KhudaBukhsh. 2024. Infrastructure ombudsman: Mining future failure concerns from structural disaster response. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 4664–4673, New York, NY, USA. Association for Computing Machinery.

Jane Arleth dela Cruz, Iris Hendrickx, and Martha Larson. 2025. Improving large language model confidence estimates using extractive rationales for classification. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 549–560, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Jane Arleth Dela Cruz, Aleksei Simonov, Iris Hendrickx, and Jurjen Wagemaker. 2026. Extracting the 'why': Flood cause annotation and LLM benchmarking from multi-platform crisis data. *Proceedings of the 23rd International Conference on Information Systems for Crisis Response and Management*.

Hieu Minh Duong, Long Nguyen, Eugene Levin, and Todd Gary. 2025. Cast: Causal discovery via spatio-temporal graphs in disaster tweets. In *2025 IEEE International Conference on Big Data (BigData)*, pages 5603–5612.

Shengnan Fu, Heng Lyu, Ze Wang, Xin Hao, and Chi Zhang. 2022. Extracting historical flood locations from news media data by the named entity recognition (ner) model to assess urban flood susceptibility. *Journal of Hydrology*, 612:128312.

Paulina Garcia-Corral, Hannah Dorothy Bechara, Krishnamoorthy Manohara, and Slava Jankin. 2025. Causally fooled in the name of being honest? evaluating causal extraction in LLMs for political text. In *Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops*, pages 36–53, Hannover, Germany. HsH Applied Academics.

Muhammad Imran, Abdul Wahab Ziaullah, Kai Chen, and Ferda Ofli. 2025. Evaluating robustness of llms on crisis-related microblogs across events, information types, and linguistic features. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 5117–5126, New York, NY, USA. Association for Computing Machinery.

Zhijing Jin, Jiarui Liu, Zhiheng LYU, Spencer Poff, Mrinmaya Sachan, Rada Mihalcea, Mona T. Diab, and Bernhard Schölkopf. 2024. Can large language models infer causation from correlation? In *The*

Twelfth International Conference on Learning Representations.

Emre Kiciman, Robert Osazuwa Ness, Amit Sharma, and Chenhao Tan. 2024. Causal reasoning and large language models: Opening a new frontier for causality. *Transactions on Machine Learning Research*. Featured Certification.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

Shinji Koju, Koichi Takeuchi, Akihiro Watanabe, Takahiro Hirayama, and Hiroyuki Nakao. 2024. Estimating task priority in japanese disaster chronology logs. In *Proceedings of the 2023 7th International Conference on Natural Language Processing and Information Retrieval*, NLPIR '23, page 304–309, New York, NY, USA. Association for Computing Machinery.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Zhenyu Lei, Yushun Dong, Weiyu Li, Rong Ding, Qi R. Wang, and Jundong Li. 2025. Harnessing large language models for disaster management: A survey. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14528–14551, Vienna, Austria. Association for Computational Linguistics.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Albert Lu, Hongxin Zhang, Yanzhe Zhang, Xuezhi Wang, and Diyi Yang. 2023. Bounding the capabilities of large language models in open text generation with prompt constraints. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1982–2008, Dubrovnik, Croatia. Association for Computational Linguistics.

Klaus-Michael Lux, Maya Sappelli, and Martha Larson. 2020. Truth or error? towards systematic analysis of factual errors in abstractive summaries. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 1–10, Online. Association for Computational Linguistics.

Emma McDaniel, Samuel Scheele, and Jeffrey Liu. 2024. Zero-shot classification of crisis tweets using instruction-finetuned large language models. *2024 IEEE International Humanitarian Technologies Conference (IHTC)*, pages 1–7.

Mistral AI Team. 2024. Mistral nemo. `https://mistral.ai/news/mistral-nemo`.

Ayoub Nainia, Régine Vignes-Lebbe, Hajar Mousannif, and Jihad Zahir. 2025. Beyond BLEU: Ethical risks of misleading evaluation in domain-specific QA with LLMs. In *Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models*, pages 77–86, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Roman Nekrasov, Stefano Fossati, Indika Kumara, Damian Andrew Tamburri, and Willem-Jan van den Heuvel. 2026. Iac generation with llms: An error taxonomy and a study on configuration knowledge injection. *ACM Trans. Softw. Eng. Methodol.*

OpenAI. 2024. Gpt-4o mini: advancing cost-efficient intelligence.

OpenAI. 2025. Gpt-5 system card.

David Otero, Javier Parapar, and Álvaro Barreiro. 2025. Limitations of automatic relevance assessments with large language models for fair and reliable retrieval evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, page 2545–2549, New York, NY, USA. Association for Computing Machinery.

H. Peng and K. Zhu. 2025. Far-am: A hybrid attention framework for fire cause classification. *PLOS ONE*, 20(10):e0333131.

Jiale Qian, Yunyan Du, Fuyuan Liang, Jiawei Yi, Nan Wang, Wenna Tu, Sheng Huang, Tao Pei, and Ting Ma. 2024. Quantifying urban linguistic diversity related to rainfall and flood across china with social media data. *ISPRS International Journal of Geo-Information*, 13(3):92.

Akond Rahman, Effat Farhana, Chris Parnin, and Laurie Williams. 2020. Gang of eight: A defect taxonomy for infrastructure as code scripts. In *2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE)*, pages 752–764.

Mayank Singh, Vikas Yadav, and Eduardo Blanco. 2026. Error taxonomy-guided prompt optimization. *Preprint*, arXiv:2602.00997.

Mai Skjott Linneberg and Steffen Korsgaard. 2019. Coding qualitative data: a synthesis guiding the novice. *Qualitative Research Journal*, 19(3):259–270.

Torgrim Solstad and Oliver Bott. 2017. Causality and causal reasoning in natural language. In *The Oxford Handbook of Causal Reasoning*. Oxford University Press.

Soudabeh Taghian Dinani, Doina Caragea, and Nikesh Gyawali. 2024. Disaster tweet classification using fine-tuned deep learning models versus zero and few-shot large language models. In *Data Management Technologies and Applications*, pages 73–94, Cham. Springer Nature Switzerland.

Zhijie Wang, Zijie Zhou, Da Song, Yuheng Huang, Shengmai Chen, Lei Ma, and Tianyi Zhang. 2025. Towards understanding the characteristics of code generation errors made by large language models.

In *Proceedings of the IEEE/ACM 47th International Conference on Software Engineering*, ICSE '25, page 2587–2599. IEEE Press.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Fengyi Xu, Jun Ma, Nan Li, and Jack C.P. Cheng. 2025. Large language model applications in disaster management: An interdisciplinary review. *International Journal of Disaster Risk Reduction*, 127:105642.

Wenda Xu, Yi-Lin Tuan, Yujie Lu, Michael Saxon, Lei Li, and William Yang Wang. 2022. Not all errors are equal: Learning text generation metrics using stratified error synthesis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6559–6574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: evaluating generated text as text generation. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Dataset & Models

Dataset. The dataset used for our experiments are from 20 different flooding events, between 2022-2025. Each flooding event is based on the location and time of the flooding event, and contains 20 to 100 documents from X (formerly Twitter), News, YouTube captions, and Bluesky posts. Each document consists of up to a maximum 500 words (texts of more than 500 words were truncated). We define causal information as information describing why the flooding event occurred and the contribution factors to the event. The annotation excluded information such as flood impacts, flood types, repairs and cleanup, future plans, general weather information, and Forecast. Table 1 contains the data split.

Model	Type	Source (OpenAI/Huggingface)
gpt-4o-mini	closed	gpt-4o-mini-2024-07-18
gpt-5-mini	closed	gpt-5-mini-2025-08-07
Llama 3.1 - 8B Instruct	open	meta-Llama/Llama-3.1-8B-Instruct
Mistral NeMo	open	mistralai/Mistral-NeMo-Base-2407

Table 5: Information of evaluated and analyzed LLMs

Models. Table 5 contains the information about the versions of the four LLMs we evaluated and analyzed. All three models, except for GPT 5-mini, were set at temperature 0.0 to be deterministic. For GPT 5-mini, we set the reasoning effort to "low".

B Prompts

Table 6 contains the flood cause document classification prompts, while Table 7 contains the sentence extraction prompts.

C Additional Results

We present the error taxonomy overview in Table 8. Table 9 shows the performance of the taxonomy-guided prompts (Def, Ex, Comb) on the flood cause sentence extraction task.

Type	Prompt
Baseline (BASE)	You are analyzing documents of flooding events. Does the following document contain information about the flood cause(s)? Answer only: Yes No
Definition (DEF)	You are analyzing documents of flooding events. Does the following document contain information about the flood cause(s)? A flood cause explains why the flooding occurred (natural or human-related) or describes the contributing factors to the flood. The cause must be an objective statement. If there is at least one cause, return Yes. Answer only: Yes No
Exclusion (EX)	You are analyzing documents of flooding events. Does the following document contain explicit information about the flood cause(s)? The following are NOT flood causes: - flood impacts and aftermath - background context - opinions and speculation - weather Forecast and flood warnings (even if they contain causal language) If the document contains 1 or more flood causes, even if any information from the list above is also present: - Return only Yes Answer only: Yes No
Combined (COMB)	You are analyzing documents of flooding events. Does the following document contain explicit information about the flood cause(s)? A flood cause explains why the flooding occurred (natural or human-related) or describes the contributing factors to the flood. The cause must be an objective statement. The following are NOT flood causes: - flood impacts and aftermath - background context - opinions and speculation - weather Forecast and flood warnings (even if they contain causal language) If the document contains 1 or more flood causes, even if any information from the list above is also present: - Return only Yes
Rationale-generation with Prediction (RAT)	Provide your best guess, and probability that it is correct (0.0 to 1.0) for the following task.Respond with only the rationale, the label, and probability. Do not modify or explain. For example: Rationale: <snippet found in input that justifies your classification, copy in verbatim> Label: <most likely label> Probability: <the probability between 0.0 to 1.0 that your classification is correct, without any extra commentary whatsoever; just the probability.> The task is: You are analyzing documents of flooding events. Based on the document's content, identify the corresponding word(s) found in the document that justify the classification as rationale. Your chosen rationale must be a snippet of the document. Copy in verbatim. Do not modify. Assign a label to whether or not the document contains a flood cause that best fits the rationale provided. Does the following document contain information about the flood cause(s)? Labels: Yes No
Few-Shot (FS)	You are analyzing documents of flooding events. Does the following document contain information about the flood cause(s)? Answer only: Yes No #Examples Document: A Tesla was spotted driving through floodwaters past another vehicle that had broken down in San Diego. At the city's airport, 2.73 inches were measured on Monday, #Flood #Floodwater #Rainfall #Heavyrain #Tesla #Sandiego #driving <URL> Answer: No Document: Cold lava floods from Mt. Marapi in Indonesia's West Sumatra province have killed 50, left 27 missing, and displaced nearly 3,400 people. Search and rescue efforts continue amid ongoing rain. #Indonesia <URL> Answer: No Document: Can someone give me a quick and dirty on why San Diego is flooding? Is it just because they've overdeveloped the land and now when it rains it doesn't drain? I know it's above sea level even though not by much Answer: No Document: Torrential rains today caused the overflow of the Barranco de la Presa in #Manises, Valencian Community, #Spain Several neighbors had to be rescued by emergency teams. The dam accumulated more than 130 liters x m².(Nov 12, 2022): Weather Orihuela #news #Flood <URL> Answer: Yes Document: SEPA Coastal Flood Warning Wednesday To Friday On Wednesday, localised coastal flooding impacts are possible in the Western Isles, the southwest and Moray due to high water levels. Localised coastal flooding impacts are more likely from Thursday around much of the coast of Scotland. On Wednesday and Thursday localised flooding impacts from rivers and surface water are possible in southern, central and north-eastern areas due to periods of heavy rain moving north over the country. Further flooding impacts from rivers and surface water are possible in the southwest of Scotland on Friday." For all areas go to <URL> Answer: No

Table 6: Flood cause document classification prompts.

Type	Prompt
Baseline	You are analyzing documents of flooding events. The following document was classified as containing information about the cause of the flood. Identify the sentences in the document that contain information about the flood cause(s). The document is split into numbered sentences. Return the sentence numbers as a list. If no sentences apply, return []. Answer only: [list of flood cause sentence numbers]
Definition	You are analyzing documents of flooding events. The following document was classified as containing information about the cause of the flood. Identify the sentences in the document that contain explicit information about the flood cause(s). A flood cause explains why the flooding occurred (natural or human-related) or describes the contributing factors of the flood. The cause must be an objective statement. The document is split into numbered sentences. Return the sentence numbers as a list. If no sentences apply, return []. Answer only: [list of flood cause sentence numbers]
Exclusion	You are analyzing documents of flooding events. The following document was classified as containing information about the cause of the flood. Identify the sentences in the document that explicitly contain information about the flood cause(s). Do not consider sentences that only contain: - background context - flood impacts and aftermath - weather Forecast and flood warnings - opinions and speculations The document is split into numbered sentences. Return the sentence numbers as a list. If no sentences apply, return []. Answer only: [list of flood cause sentence numbers]
Combined	You are analyzing documents of flooding events. The following document was classified as containing information about the cause of the flood. Identify the sentences in the document that explicitly contain information about the flood cause(s). A flood cause explains why the flooding occurred (natural or human-related). The cause must be an objective statement. Do not consider sentences that only contain: - background context - flood impacts and aftermath - weather Forecast and flood warnings - opinions and speculations The document is split into numbered sentences. Return the sentence numbers as a list. If no sentences apply, return []. Answer only: [list of flood cause sentence numbers]

Table 7: Flood cause sentence extraction prompts.

Error Category	Definition	Example
Impact Information and Aftermath	Direct consequences of the flood event. This includes impact information about affected people, infrastructure damage and flooding aftermath information.	Residents 'evacuated' from rural homes near Chester due to flooding
Background Context	General context about the flooding situation and location but no information about the direct cause.	Hail storm and a record amount of precipitation and lightning in Valencia city <URL>
Subjective and Unverified Causality	Opinionated, speculative, and/or implicit causal statements. This includes political blaming, sarcasm, and framing of hypothetical questions.	@USER1 @USER2 And the cause of all the bad weather, the Indian Monsoon in South Florida, overload of emergency systems, school busses not in route, the floods, car breakdowns and soil over-saturation, you still think weather manipulation and SRM is good for us, you think that this is all good? <URL>.
Weather Forecast and Warnings	Weather forecast of heavy rain and flooding that is yet to happen. This includes flood warnings, advice, and precaution.	Heavy rainfall today may cause some flash flooding across Scotland. See here for more information and advice - <URL>
Unrelated Mentions	Metaphorical and out of the natural hazard context use of the word,"Flood".	Eight people die in the London Beer Flood, an accident at Meux & Co's Horse Shoe Brewery, London.

Table 8: Error Taxonomy of False Positives for the Document-level Flood Cause Classification

Flood Cause Sentence Extraction

Model	Prompt	P	R	PR-Gap ↓	F0.5-score
GPT 4o-mini	BASE	73	90	17	76
	DEF	72	91	19	75
	EX	79	82	**4**	79
	COMB	79	84	5	80
GPT 5-mini	BASE	72	96	24	76
	DEF	71	97	26	75
	EX	76	95	19	79
	COMB	76	95	**18**	79
Llama 3.1	BASE	58	93	35	62
	DEF	60	91	31	64
	EX	61	85	23	65
	Comb	68	79	**11**	70
Mistral Nemo	BASE	68	76	9	69
	DEF	66	78	12	68
	EX	71	73	**2**	71
	COMB	70	73	3	71

Table 9: Performance of LLMs in the flood cause sentence extraction task using our prompt constraints on the validation set. Performance is measured on Accuracy (A), Precision (P), Recall (R), F0.5-score, and the Precision-Recall Gap (PR-Gap).

Document	Error Category	Def	Ex	Comb
Only home 4 days. Since then everywhere has flooded and it's been blowing a hoolie for 2 days. Nothing like west of Ireland weather to make you wonder if it is indeed possible ☼	Background Context	✔	✔	✔
Potential #TropicalCyclone life-threatening flash flooding expected in #Honduras later this week and over the weekend...#Nineteen Pressure: 1005 mb, Wind Speed: 30 mph <URL> <URL>	Forecast and Warning	✘	✔	✔
#Honduras battles severe flooding from #TropicalStormSara, displacing over 1,000 people as it threatens #Belize and #Guatemala. #HondurasFlooding #TropicalStorm #heavyrain #CentralAmerica #SevereWeather #Flooding #HurricaneSeason #TheDailyIntake	Impact and Aftermath	✘	✘	✔
The rate of flooding in Lagos is alarming. It questions or preparedness towards climate-change adaptation. We are still unable to meet up with resilience.	Subjective and Unverified	✔	✔	✔
Is there a relationship between #climatechange and conflict? On the latest episode of Lagos to Mombasa, @USER joins host @USER to discuss the impact of rising temperatures, extreme droughts, & floods on competition for resources. Tune in <URL>	Unrelated Mention	✔	✔	✔

Table 10: Examples of false positives that were corrected by the prompt interventions. The examples were predictions from GPT 4o-mini prompt interventions. The ✔ mark indicates the prompt intervention was able to suppress the false positive while ✘ mark indicates the intervention was not able to suppress it. Def stands for Definition-Aware prompt, Ex for exclusion-aware and Comb for the combined (definition and exclusion).

GeNERscore: Semantic Alignment Metrics for Generative Named Entity Recognition

Gianmarco Pappacoda[1], Federico Ruggeri[1], Paolo Torroni[1]

[1]DISI, University of Bologna
{gianmarco.pappacoda, federico.ruggeri6, p.torroni}@unibo.it

Abstract

Traditional Named Entity Recognition (NER) metrics rely on syntactic span and type matching. While this may be appropriate if NER is framed as a sequence labeling task, the advent of generative models allows for alternative approaches, where model outputs are not extracted, but generated. In this perspective, generative NER calls for more semantically-aware scoring mechanisms. Generative models may produce valid aliases, paraphrases, or type variants that differ from the annotated gold, while also producing unsupported predictions. We propose a taxonomy of such cases and introduce an embedding-based parametric method to evaluate generative NER based on three metrics: *soft precision*, *soft recall* and *unsupported rate*. We validate this framework using standard and noisy datasets, across state-of-the-art systems such as GPT and GLiNER alongside older encoder-based models. We find that our metrics mostly agree with traditional ones on better-established, less noisy benchmarks, but differ in more recent, noisier ones, and can capture edge cases that arise from free-form generation.

1 Introduction

Named Entity Recognition (NER) aims at recognizing entities, each described by a *mention*, i.e., a surface-form span of text, and a *type*, i.e., the semantic class assigned to it. NER is typically evaluated by comparing the entity spans extracted by a model against gold-standard annotations. Traditionally, a prediction is correct only when both its boundary and type match the gold annotation exactly, a convention that runs from the MUC evaluations through CoNLL-2003 (Chinchor, 1995; Tjong Kim Sang and De Meulder, 2003). To relax the rigidity of exact-matching, SemEval-2013 introduced softer views of correctness, including partial and type matching, which separate boundary errors from type errors (Segura-Bedmar et al., 2013). These views recognize near misses, but they remain surface-bound: they decide correctness through span identity or overlap rather than semantic equivalence. Accordingly, they cannot credit aliases, paraphrases, or descriptive mentions that do not overlap the gold span, and may overcredit predictions with a high overlap with the gold span but no semantic correspondence with it.

Generative models introduce further complications. Unlike sequence taggers, generative models are not restricted to copying input spans. Instead, they may normalize or paraphrase a mention, or choose a different label name. These could be useful, welcome features, but they harm the existing evaluation methods. Consider a model that predicts *JFK* for the gold entity *John F. Kennedy*. They name the same person, but share no contiguous substring, so any kind of syntactic matching, exact or partial, would score the prediction as wrong, which may be too penalizing. To complicate things further, generative models are also prone to hallucinations, therefore the generated text may be unrelated to the source. String matching cannot distinguish between a semantically sound rephrase and hallucinated text. These mismatches are most visible on noisy, user-generated text, where entities typically appear under many aliases or with rare and unstable surface forms (Baldwin et al., 2015; Derczynski et al., 2017). We posit that noisy text should not be treated as a special case, but as the general form NER inputs may take, and that NER evaluation should reflect this setting.

We propose Generative NER score (**GeNERscore**), an embedding-based and entity-level view of NER evaluation, inspired by semantic metrics for text generation that complement exact overlap with contextual similarity (Zhang et al., 2020). Rather than matching strings, **GeNERscore** represents gold and predicted typed entities in the same embedding space, matches them with a configurable rule, and

Proceedings of the 11th Workshop on Natural User-generated Text, pages 117–140
October 28, 2026 ©2026 Association for Computational Linguistics

scores such an alignment with soft analogues of precision and recall. Alongside these metrics, we also propose an indicator of over-generated entities.

We make three major contributions:

- We articulate a taxonomy of evaluation mismatches for generative NER.
- We introduce **GeNERscore**, a semantic alignment framework that compares predicted and gold entities through embedding similarity and configurable assignment algorithms.
- We evaluate **GeNERscore** on four datasets and three model families, showing that it agrees with exact and partial F1 on discriminative models and recovers credit withheld by string matching on generative ones.

2 Motivation

To motivate our contribution, we illustrate six non-mutually exclusive categories of phenomena where string-based evaluation fails to account for meaning equivalence. For each category, we report one example, taken from predictions that strict matching scored as wrong across our datasets and models (Section 5) but could be acceptable, semantically speaking. These examples are illustrative and are not claimed to be statistically representative; they are chosen to show generative models' behaviour and how current evaluation can be overly penalizing for them.

> *"For Taiwan firms in Dongguan, investing in mainland **China** is a road of no return."*
> — OntoNotes 5 #6454

Alias and surface-form variation. The gold annotates `China`/GPE, while both GPT 5.4-mini and GLiNER2 predict `mainland China`/GPE. Exact matching scores this as wrong and partial overlap recovers some credit, yet the prediction names the same place. A semantically-aware metric should reward it.

> *"24 days until **@Cyberpunk 2077@**** comes out. I am ready baby !"*
> — TweetNER7 #476

Type disagreement. The span matches exactly and only the type label differs. The gold annotates `@Cyberpunk 2077@`/product, while GPT 5.4-mini and GLiNER models predict `Cyberpunk 2077`/creative_work, two categories that overlap heavily, as in this span, which qualifies as both. Partial F1 collapses to zero on this span, even though the labels are similar. A semantically-aware metric should take into account type similarity too, without over-penalizing near-correct predictions.

> *"Give a Candlestick **Park card for the memories!** #SFGiants {{URL}} via {@Etsy@}"* — TweetNER7 #2245

Semantic paraphrase. The gold annotates `Park card for the memories`/product, while GLiNER model predicts `Candlestick Park card`/product. The two annotations name the same item, but the surface forms differ enough that exact match and most partial matching methods score it as wrong despite the equivalent meaning. A semantically-aware metric should account for this equivalence.

> *"Result of an **English F.A. Challenge**"*
> — CoNLL-2003 #139

Partial-overlap over-credit. The gold marks `English F.A. Challenge`/MISC while the encoder-based model predicts `English`/MISC, dropping most of the name. While they differ in meaning, a partial match would reward this prediction. A semantically-aware metric should credit the prediction proportionally to its semantic equivalence to the gold annotation. In this case, penalize entirely as they differ.

> *"... WE LOVE YOU SOHEL **#HBDIsmartSohel #HBDSyedSohelRyan**"*
> — TweetNER7 #2711

Annotation incompleteness. The gold annotates only the two hashtags as person mentions, whereas GPT 5.4-mini model also predicts `SOHEL`/person, a named individual the gold omits. The unannotated entity counts as a false positive against string-based metrics and penalizes precision.

> *"**Jill Biden** Leaves the **White House** in **Green Jewel-Toned Jacket** & Her Favorite **Boot Trend** {{URL}} via **{@Yahoo@}** Hey **{@MELANIA TRUMP@}**,did you see what **Dr. Biden** was wearing the other day ?"*
> — TweetNER7 #2414

Hallucination. Among several correct predictions, GPT 5.4-mini also generates `Whatsup?`no/creative_work, a span that does not appear in

the input text and matches no gold entity. In a generative setting, a semantically-aware metric should be able to tell apart correct extractions (i.e., aliases, paraphrases and omissions in the gold) from true hallucinations.

3 Related Work

Traditionally, NER has been evaluated with precision, recall, and F1 under exact span and label match, where a prediction is correct only if both its boundary and type are identical to the gold annotation (Chinchor, 1995; Tjong Kim Sang and De Meulder, 2003). SemEval-2013 Task 9.1 softened this setting by proposing multiple views of correctness, including strict, exact, partial, and type matching (Segura-Bedmar et al., 2013).

Noisy user-generated text. User-generated text has made these limitations especially visible (Baldwin et al., 2015; Strauss et al., 2016). Derczynski et al. (2017) shifted attention to novel and emerging entities, noting the recall difficulties caused by rare and unstable surface forms, while recent work continues this concern through noise analyses for NER (Bhadauria et al., 2024) and Large Language Model (LLM)-based extraction from informal text (Popovic et al., 2024). Our work follows this line, but treats noisy text as one instance and a symptom of a broader evaluation problem.

Relaxing exact scoring. Several lines of work relax exact scoring using external structures. Hierarchical precision/recall/F1 gives partial credit when predicted and gold labels are close in a taxonomy (Kiritchenko et al., 2006), while canonicalization approaches resolve entities to a common name through span-overlap scoring, type weights, entity linking, or coreference (Hachey et al., 2014; Agarwal et al., 2019; Bast et al., 2023). Yamada et al. (2015) and Limsopatham and Collier (2016) explored entity linking and neural sequence models. Generative linking systems such as mGENRE predict canonical entity names autoregressively, but require a Knowledge Base (KB) and a linkable target (De Cao et al., 2022). These approaches show that exact-match F1 is often too strict, but they typically require ontology-aware resources, alias tables, or a fixed KB, mitigating closed-world variation rather than addressing open-ended generative outputs.

LLM-based entity matching and linking. Recent work has applied generative LLMs to entity matching and entity linking. Peeters and Bizer (2023); Peeters et al. (2025) show that GPT can perform competitive in zero-shot and few-shot entity matching with far fewer labeled examples than a fine-tuned encoder. Wang et al. (2025b) move beyond binary decisions by prompting LLMs to compare or select among multiple candidate records. For entity linking, Ding et al. (2024) constrain an LLM to choose among retrieved candidates, achieving comparable performance to supervised model. However, these methods are susceptible to LLM selection and prompt variation, leading to more instability during matching/linking.

Generative-based NER. Recent work in generative relation extraction argues that reference-based precision and recall are not sufficient when LLMs can produce valid outputs that differ from the annotation surface form. Jiang et al. (2024) make this point explicitly in GenRES, which augments standard scores with dimensions such as factualness and completeness. From a more general structured-prediction perspective, Chen et al. (2023) argue that evaluation design strongly shapes what systems are rewarded for producing. In generative NER specifically, systems such as GPT-NER reformulate extraction as text generation and highlight hallucination control as a central problem (Wang et al., 2025a). Recent surveys document the rapid adoption of generative Information Extraction (IE) pipelines across NER, relation extraction, and event extraction (Xu et al., 2024). More broadly, semantic generation metrics such as BERTScore (Zhang et al., 2020) show the value of replacing string match overlap with contextual similarity while keeping the precision/recall intuition intact. Our setting differs from pure free-form generation (e.g., Machine Translation), but the underlying evaluation problems are shared.

Annotation errors. A growing body of work shows that gold NER and IE annotations contain omissions and outright mistakes, which become more apparent as the model quality increases. For instance, Reiss et al. (2020) identify over 1.3k incorrect labels in CoNLL-2003 and note that the number of test-set errors is comparable to the remaining errors of strong models; Rueda et al. (2024) subsequently introduce CoNLL# as a corrected test set for lower-noise evaluation. For this reason, researchers treat partially annotated or unlabeled entities as a central learning problem rather than as negligible noise (Mayhew et al., 2019; Peng

et al., 2019; Li et al., 2021). Similar observations hold beyond NER: Alt et al. (2020) show that relation extraction performance on TACRED is substantially distorted by annotation error, and Nejadgholi et al. (2020) report that for overlapping same-label span mismatches in medical NER, expert users accept or partially accept the large majority of cases. This literature supports a key premise of our work: in tasks like generative NER, a false positive may be an annotation error or a model error, such as a hallucination, but it may also be an alternative mention or a contextually relevant prediction.

Our work is close in spirit to the semantic and canonicalization approaches, but differs in two ways. First, we target generative NER directly, where the output may be a plausible mention or type that is not ontology-normalized and may not overlap with the gold span. Second, instead of collapsing everything onto a single metric, we provide a separate metric that shows how many over-generated predictions are unsupported by the gold annotation. Like ontology-aware and linking-based approaches, **GeNERscore** credits semantic equivalence, but unlike them, it does not require a reference glossary or knowledge base.

4 GeNERscore

We describe **GeNERscore**, a semantically-aware evaluation framework for generative and open-vocabulary NER that operates on the embedding space rather than relying on string matching.

4.1 Motivation

Let $\bar{E} = \{\bar{e}_1, \ldots, \bar{e}_N\}$ denote the gold typed entities and $\hat{E} = \{\hat{e}_1, \ldots, \hat{e}_M\}$ the predicted typed entities. Under strict evaluation, a prediction is correct only if it exactly matches a gold entity in both mention and type. Writing $\mathbf{1}[\bar{e}_i = \hat{e}_j]$ for this exact-match indicator, the standard entity-level scores are as follows.

$$\textit{Exact Precision} = \frac{1}{M}\sum_{j=1}^{M}\max_{i=1}^{N}\mathbf{1}[\bar{e}_i = \hat{e}_j], \quad (1)$$

$$\textit{Exact Recall} = \frac{1}{N}\sum_{i=1}^{N}\max_{j=1}^{M}\mathbf{1}[\bar{e}_i = \hat{e}_j] \quad (2)$$

These metrics are easy to interpret and are applied when span-faithful extraction and type identity are required. However, they also come with an important consideration: a prediction either gets full credit or none.

One relaxation is partial matching, which could be implemented in multiple ways based on substring or span overlap. It gives credit to near misses due to boundary drift, while still limited to surface-form similarity. It helps with cases like *Apple* versus *Apple Inc.*, but not with aliases, descriptive mentions, or semantically equivalent references. In summary, partial matching softens exact scoring without distinguishing semantic recovery from partial overlap over-credit. Such considerations might not be ideal for generative and open-vocabulary NER, where the evaluation focuses on whether predicted typed entities belong to the same semantic frame defined by the domain of interest rather than on strict string matching (§2), including cases such as paraphrases, hypernyms, acronyms, and, more in general, semantic relatedness.

We design **GeNERscore** to account for such cases by satisfying the following desiderata via the replacement of string matching with embedding-based semantic similarity. First, it rewards semantic similarity beyond lexical identity, to accommodate aliases, abbreviations, and spelling variants. Second, it handles type mismatch beyond exact match, to also reward types' semantic similarity. Third, it still penalizes incompatible types and wrong boundaries, such that mention or type overlap alone are not sufficient for correctness. Fourth, it exposes over-generated entities via a separate score.

4.2 Method

GeNERscore operates by computing the similarity matrix $S \in \mathbb{R}^{N\times M}$, where each entry S_{ij} quantifies how semantically close gold entity $\bar{e}_i$ is to predicted entity $\hat{e}_j$. Formally, we assume a pre-trained text embedding model ϕ : String $\to \mathbb{R}^D$ that maps arbitrary strings to D-dimensional embeddings. We denote a typed entity e as a pair $e = (e^{\text{txt}}, e^{\text{typ}})$, where e^{txt} is the mention string and e^{typ} is the type label. **GeNERscore** is therefore entity-level: it compares entity strings, rather than considering per-token scoring. We do so to account for artifacts a generative model might produce, such as normalized, paraphrased, or descriptive mentions that are not a token-by-token copy of the input. Given embedding model ϕ and typed entity e, we define its embedding as follows:

$$\begin{aligned}\mathbf{e} &= (\mathbf{e}^{\text{txt}}, \mathbf{e}^{\text{typ}}),\\ \mathbf{e}^{\text{txt}} &= \phi(e^{\text{txt}}) \in \mathbb{R}^D,\\ \mathbf{e}^{\text{typ}} &= \phi(e^{\text{typ}}) \in \mathbb{R}^D\end{aligned} \quad (3)$$

We represent each typed entity e with separate mention and type embeddings. The mention embedding captures referential similarity between surface forms, while the type embedding captures compatibility between labels. For both embeddings, we embed the given raw string (e.g., PER for types). We apply no canonicalization or mapping to keep the metric ontology-agnostic: it requires no per-dataset label dictionary and allows for free-form or previously unseen labels.

Conceptually, S is an entity-level analogue of semantic generation metrics such as BERTScore (Zhang et al., 2020). The main difference is that the pairwise comparison is between typed spans rather than tokens.

Type-aware Cosine Similarity. Given gold and predicted entity embeddings, we define S by computing a similarity function $s(\cdot, \cdot)$ of each $(\bar{\mathbf{e}}, \hat{\mathbf{e}})$ pair as follows.

$$S_{ij} = s(\bar{\mathbf{e}}_i^{\text{txt}}, \hat{\mathbf{e}}_j^{\text{txt}}) \cdot s(\bar{\mathbf{e}}_i^{\text{typ}}, \hat{\mathbf{e}}_j^{\text{typ}}) \tag{4}$$

where

$$s(\bar{\mathbf{e}}_i^{\text{txt}}, \hat{\mathbf{e}}_j^{\text{txt}}) = max\left(0, \text{cos\_sim}(\bar{\mathbf{e}}_i^{\text{txt}}, \hat{\mathbf{e}}_j^{\text{txt}})\right), \tag{5}$$

$$s(\bar{\mathbf{e}}_i^{\text{typ}}, \hat{\mathbf{e}}_j^{\text{typ}}) = max\left(0, \text{cos\_sim}(\bar{\mathbf{e}}_i^{\text{typ}}, \hat{\mathbf{e}}_j^{\text{typ}})\right) \tag{6}$$

We keep the two comparisons separate and combine them multiplicatively so that credit requires *joint* compatibility: a low score in either embedding suppresses the product, whereas a sum or average would let a strong mention match mask an incompatible type. Each similarity is clipped at zero before being multiplied, so a negative (incompatible) mention or type similarity contributes no credit, rather than flipping the sign of the product. This bounds every entry $S_{ij} \in [0, 1]$ and keeps the score monotone in both embeddings.

Interpretation. Each row i of S shows how well gold entity $\bar{e}_i$ is matched by every prediction; the row maximum identifies its best candidate recovery. Each column j shows how well the predicted entity $\hat{e}_j$ aligns with gold annotations; the column maximum indicates its strongest semantic support. High values suggest referential and type compatibility, whereas uniformly low columns indicate likely unsupported predictions. This row/column duality allows S to expose entity-level alignment and to support both recall-like and precision-like views.

4.3 Assignment

The similarity matrix alone does not specify which predictions should count as matches for gold entities. We therefore define an assignment set $\mathcal{A}$ over pairs (i, j) and discuss three algorithms to define $\mathcal{A}$.

4.3.1 Hungarian Assignment

The Hungarian algorithm produces a globally optimal one-to-one alignment in polynomial time. (Kuhn, 1955) Polynomial complexity is important as each matrix S instance is composed of N rows and M columns, thus potentially quadratic. Finding the optimal alignment in time may be difficult for large documents. The Hungarian algorithm solves this issue.

$$\mathcal{A}_H = \arg\max_{\mathcal{A}} \sum_{(i,j)\in\mathcal{A}} S_{ij} \tag{7}$$

subject to the constraint that each gold entity and each prediction participates in at most one pair. This provides a maximum-weight matching, but it cannot express cases where one entity is split across several predicted mentions. It is therefore too restrictive as a standardized assignment rule for generative NER: aliases, expanded descriptions, split and merged entities may produce many-to-many alignments.

4.3.2 Threshold Assignment

Threshold assignment relaxes one-to-one matching by selecting every pair above a user-specified threshold $\tau \in [0, 1]$.

$$\mathcal{A}_\tau = \left\{(i, j) : S_{ij} \geq \tau\right\} \tag{8}$$

This many-to-many strategy is better suited to boundary ambiguity and split or merged predictions, but it depends directly on threshold calibration. As τ rises, low-confidence but still best-available correspondences are removed entirely. This can lower the recall-oriented score even when every gold entity has a plausible best prediction, because threshold assignment does not guarantee that each gold item receives any partner.

4.3.3 Hybrid Assignment

A hybrid strategy that combines the Hungarian assignment with threshold-based expansion as follows.

$$\mathcal{A}_{\text{hybrid}} = \mathcal{A}_H \cup \mathcal{A}_\tau \tag{9}$$

The result preserves the best one-to-one backbone while still admitting additional semantically plausible matches. The Hungarian component guarantees entity coverage up to the available number of predictions, while the threshold component maintains the many-to-many assignment needed for the reasons discussed above.

4.4 Evaluation Metrics

Given S and an assignment $\mathcal{A}$, **GeNERscore** comprises three complementary metrics. The first two are soft analogues of recall and precision. The third measures how much of a model's prediction is not supported by gold entities.

4.4.1 Soft Recall

For each gold entity $\bar{e}_i$, we take the best similarity among all predictions assigned to it, or 0 if it is unmatched.

$$r_i = \begin{cases} \max\limits_{j:(i,j)\in\mathcal{A}} S_{ij} & \text{if } \exists j \text{ s.t. } (i,j) \in \mathcal{A}, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

We then define soft recall as follows.

$$\textit{Soft Recall} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{11}$$

While there may be multiple assigned predictions to a gold entity, we take the maximum similarity over assigned predictions rather than their mean. A model that generates several plausible aliases of one gold entity (e.g., *Apple*, *Apple Inc.*) is therefore not heavily penalized for possible redundancies in generated entities.

4.4.2 Soft Precision

For each prediction $\hat{e}_j$, we take the mean similarity across its assigned gold assignments when it is matched.[1] Otherwise, in the case $\hat{e}_j$ is unmatched, we consider the best similarity score across gold annotations.

$$c_j = \begin{cases} \dfrac{1}{|C_j|}\displaystyle\sum_{i:(i,j)\in\mathcal{A}} S_{ij} & \text{if } C_j \neq \emptyset, \\ \max\limits_{i=1}^{N} S_{ij} & \text{otherwise} \end{cases} \tag{12}$$

where $C_j = \{i : (i,j) \in \mathcal{A}\}$ is the set of gold entities indexes assigned to prediction j. We define soft precision as follows.

$$\textit{Soft Precision} = \frac{1}{M}\sum_{j=1}^{M} c_j \tag{13}$$

We average over assigned predictions to account for the case where a prediction might be matched to more than one gold entity. By doing so, we avoid crediting a prediction solely for its best match when considering a many-to-many assignment. This design finds its motivation when considering the case of over-generation, which occurs when a model generates more entities $\hat{e}$ than the available gold annotations $\bar{e}$ (i.e., $M > N$). In such a case, soft precision weights surplus predictions proportionally to their similarity scores rather than attributing a binary penalization (i.e., either correct or incorrect). For instance, a surplus mention that is a plausible alias of a gold entity is penalized only lightly. Instead, an unmatched mention with low similarity score lowers the overall precision score.

4.4.3 Unsupported Rate

The unsupported rate quantifies the proportion of predictions that are distant from the gold entities, according to some similarity threshold. It is assignment-independent. For each prediction, we compute its best similarity to any gold entity (i.e., its *support*).

$$h_j = \max_{i=1}^{N} S_{ij} \tag{14}$$

Given a support threshold τ_u, we flag predictions with insufficient support as unsupported.

$$\textit{Unsupported Rate} = \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}\big[h_j < \tau_u\big] \tag{15}$$

It is worthwhile noticing that an unmatched prediction may be the product of a genuine hallucination or of a mention that the model correctly emits but the gold annotation fails to specify (§2). This metric should therefore be used as a way to estimate the ratio of the prediction that is over-generated or not correspondent to any gold, rather than a hallucination gauge.

5 Experimental Setup

To validate our evaluation framework, we consider two settings. In the first, we assess whether semantic scoring agrees with existing scoring methods on standard NER benchmarks, where supervised

[1] For one-to-one assignment, this reduces to the single assignment score.

discriminative models achieve near-optimal performance according to exact-match scoring. As benchmarks, we consider CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) and OntoNotes 5 (Weischedel et al., 2013). In the second setting, we stress-test the metrics to expose different kinds of divergence, seen in §2, between strict string matching and semantically-aware scoring. As benchmarks, we consider TweetNER7 (Ushio et al., 2022), a noisy dataset we deem valuable for assessing aliasing, abbreviation, annotation omission, boundary variation, and emerging surface forms. Additionally, we consider SciERC (Luan et al., 2018), a scientific-domain dataset where annotations contain specialized terminology and annotation disagreement often concerns type mismatch and technical paraphrasing rather than informal spelling or event-driven naming.

Models. We consider three model architectures: T-NER (Ushio and Camacho-Collados, 2021), GPT 5.4-mini (Singh et al., 2025)[2] and GLiNER2 (Zaratiana et al., 2025)[3]. T-NER represents task-specific token-classification systems comprising a transformer encoder and Conditional Random Field (CRF)-style decoding. We use the existing T-NER model checkpoints for CoNLL-2003[4], OntoNotes 5[5] and TweetNER7[6]. For SciERC, we train a ModernBERT (Warner et al., 2025) encoder using T-NER implementation with default hyperparameters on the training set. T-NER represents a strong baseline when trained for a target dataset. Nonetheless, the model requires re-training when the domain or the set of entity types changes. GPT 5.4-mini represents instruction-following generative extraction. It can be prompted with a new set of entity types without task-specific fine-tuning. For each dataset, we zero-shot prompt GPT 5.4-mini with the dataset's allowed entity types to return a JSON array of `{text, type}` objects, which we parse into typed entity spans (see Appendix A). GLiNER2 represents a middle ground between fixed token classifiers and fully generative LLMs: it supports open-vocabulary entity types through natural-language labels with an encoder model backbone.

[2]Accessed via the OpenAI API (`GPT-5.4-mini`); see `https://platform.openai.com/docs/models`, June 2026.

[3]`gliner2-large-v1`.

[4]`deberta-v3-large-conll2003`.

[5]`roberta-large-ontonotes5`.

[6]`roberta-large-tweetner7-all`.

Model	Strict F1 ↑	Partial F1 ↑	Soft F1 ↑	U. Rate ↓
CoNLL-2003 ($n = 3453$)				
T-NER	0.922	0.933	0.945	0.030
GLiNER2	0.355	0.380	0.592	0.470
GPT-5.4-mini	0.750	0.768	0.833	0.045
OntoNotes 5 ($n = 8262$)				
T-NER	0.931	0.952	0.955	0.028
GLiNER2	0.348	0.418	0.533	0.577
GPT-5.4-mini	0.775	0.852	0.864	0.052
TweetNER7 ($n = 2807$)				
T-NER	0.637	0.693	0.818	0.151
GLiNER2	0.263	0.324	0.581	0.516
GPT-5.4-mini	0.320	0.586	0.708	0.285
SciERC ($n = 100$)				
T-NER	0.505	0.650	0.744	0.117
GLiNER2	0.326	0.449	0.669	0.302
GPT-5.4-mini	0.324	0.422	0.640	0.283

Table 1: Results across all four datasets, macro-averaged over instances.

Evaluation. We compute **GeNERscore** metrics after parsing a model's predictions into typed entity spans. For each dataset instance with gold entity types $\bar{E}$ and predicted entity types $\hat{E}$, we perform experiments with each assignment method and different thresholds. For each experiment, we compute the similarity matrix S, the assignment $\mathcal{A}$, and the metrics. We macro-average metrics over dataset instances in a given split. We supply GPT-5.4-mini and GLiNER2 with a dataset's entity type ontology, such that gold and predicted entity types are drawn from the same vocabulary. We apply no further canonicalization to either type or mention strings before embedding them. We compute **GeNERscore** metrics using the embedding models and thresholds reported in Appendix B. We define partial matching in the following way: $\bar{e}^{txt}$ is a substring of $\hat{e}^{txt}$ or $\hat{e}^{txt}$ is a substring of $\bar{e}^{txt}$ and $\bar{e}^{typ} = \hat{e}^{typ}$. This corresponds to partial+type of Segura-Bedmar et al. (2013). Due to space limitations, we do not report all the data and figures related to all alignment algorithms, models and thresholds, but focus only on the most significant configurations.

6 Results

We perform an initial analysis on all configurations (Appendix C). We observe that the Hungarian assignment is too limiting, and threshold assignment can be too sensitive to the calibration of τ. Therefore, we focus on hybrid assignment. We also observe both in $\mathcal{A}_\tau$ and $\mathcal{A}_{hybrid}$ (§4), that *soft precision* and *soft recall* tend to meet around $\tau = 0.6$. Therefore, in Table 1, we report results using $\mathcal{A}_{hybrid}$, $\tau = \tau_u = 0.6$ and $\phi =$ Qwen3-

embedding-4B as the embedding model, since it is the largest one in our experiments. For brevity, we summarize soft metrics with their harmonic mean, denoted as *Soft F1*. We observe two patterns. First, the discriminative T-NER models are the best-performing ones on all four datasets under both strict and **GeNERscore** evaluation. This demonstrates that our soft variants do not disagree with traditional metrics on discriminative models. Second, the gap between Strict F1 and Soft F1 is especially large for GPT 5.4-mini on a noisy benchmark such as TweetNER7. In this setting, the difference between GPT 5.4-mini's and T-NER's Soft F1 is much less pronounced than the difference between their Strict F1 scores. Both GPT 5.4-mini and GLiNER2 over-generate entities, yet both gain from semantic scoring: their Soft F1 exceeds their Partial F1 on every dataset. What separates them is the Unsupported Rate. GLiNER2 leaves roughly half of its predictions unsupported on CoNLL-2003, OntoNotes 5, and TweetNER7, which hampers its Soft F1. The two models converge on SciERC, where GLiNER2's unsupported rate drops to 0.302 and its Soft F1 (0.669) slightly exceeds that of GPT 5.4-mini (0.640).

SciERC is the most challenging dataset in our experiments. In Figure 1, we report the different behaviors of our metrics when varying the assignment algorithm and thresholds. Hungarian assignment yields fixed soft precision and soft recall regardless of τ, providing the best one-to-one assignment. Threshold assignment allows calibration, but soft precision and soft recall diverge at the extremes of τ range. Hybrid assignment combines the stability of optimal matching with threshold-based filtering. We observe precision and recall curves cross near $\tau = 0.6$. The same pattern holds on the remaining datasets (Appendix C). We also verify that the choice of embedding model has little effect on the metrics (Appendix B).

7 Discussion

On CoNLL-2003 and OntoNotes 5, the supervised T-NER models have near-ceiling performance and are the best-performing models. **GeNERscore** does not over-allocate credit where exact match already works. Regarding TweetNER7 and SciERC, model performance drops compared to standard benchmarks and the strict-to-soft gap widens, most prominently for the generative model. Performance drop could be due to the fact that standard benchmarks are simple and well-curated, while TweetNER7 adds noise and SciERC adds document context and specialized terms. Another hypothesis is test-set leakage into the backbones' pretraining data.

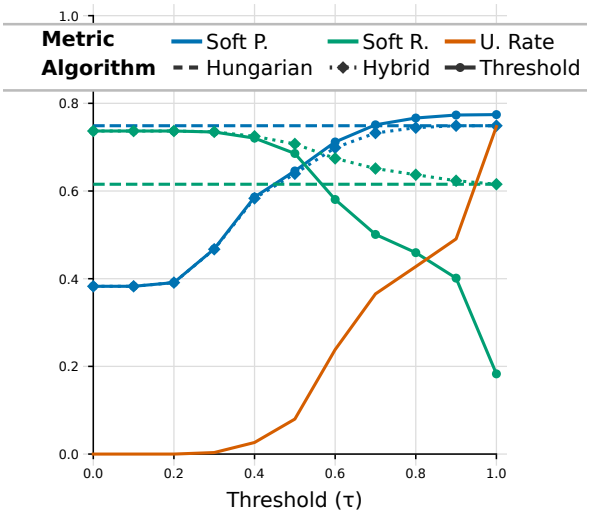


Figure 1: **GeNERscore** metrics for SciERC. Results averaged across models (T-NER, GLiNER2, GPT 5.4-mini). Thresholds $\tau = \tau_u$ for the three assignment algorithms. Hungarian assignment (dashed) is τ-independent. The unsupported rate is assignment-independent and appears as a single orange curve.

The gap in the noisier data is what the metric is meant to catch. GPT 5.4-mini produces aliases, paraphrases, and near-synonymous type labels that strict matching scores as wrong. Soft scoring recovers this credit, while the unsupported rate stays above the supervised model's, so **GeNERscore** does not reward all predictions. Soft scoring does not replace exact and partial matching, which remain the right choice when exact mention and type identity are important. **GeNERscore** is complementary and most useful for generative and open-vocabulary systems where string matching is less reliable.

8 Conclusion

Exact-match NER evaluation is well suited to closed, span-faithful extraction, but it becomes increasingly brittle when model outputs and annotations diverge syntactically. We have shown six cases in which this evaluation falls short: alias and surface-form variation, type disagreement, semantic paraphrase, partial-overlap over-credit, annotation incompleteness and hallucination. **GeNERscore** is a modular framework that evaluates NER based on the semantic alignment between predicted

and gold typed entities.

We validate the framework against four datasets and three model families. We show that it does not disagree with traditional metrics on discriminative models. On generative models, it recovers credit that existing string-matching metrics withhold. Importantly, **GeNERscore** flags overgeneration through a new metric called *unsupported rate*. We show that our metrics are largely insensitive to the embedding model used to compute them, which makes them computationally viable. By agreeing with traditional metrics in cases where they are reliable and capturing edge cases where they struggle, **GeNERscore** enables a more semantically sensitive evaluation of generative NER systems.

Limitations

Soft precision credits a surplus prediction by its best available support, so it is lenient toward overgeneration that stays close to the gold. That is what we want for generative output. When a stricter evaluation is needed, employing traditional precision is still advisable.

While we show that our metrics are robust with respect to the choice of embedding models (Appendix B), they inherit the limitations and biases of the underlying embedding model.

All our metrics rest on the assumption that a high alignment score reflects referential and type similarity. A human alignment study would test this directly. As this would entail a significant amount of work, we leave it to future works, alongside testing datasets in other languages and domains.

While LLM-based methods come with different biases and suffer from prompt sensitivity, they can provide reasoning behind a given match. A comparison with them on entity matching and entity linking could help in untangling the factors described above.

Acknowledgements

Thanks to DecisionBrain for providing computational resources to run the experiments.[7]

AI assistance. Generative AI tools were used to assist with language editing and improve the clarity of the manuscript. The authors reviewed and revised all generated suggestions and take full responsibility for the content of the paper.

[7] https://decisionbrain.com

References

Oshin Agarwal, Sanjay Subramanian, Ani Nenkova, and Dan Roth. 2019. Evaluation of named entity coreference. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–7, Minneapolis, USA. Association for Computational Linguistics.

Mohammad Kalim Akram, Saba Sturua, Nastia Havriushenko, Quentin Herreros, Michael Günther, Maximilian Werk, and Han Xiao. 2026. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*.

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135. Association for Computational Linguistics.

Hannah Bast, Maximilian Hertel, and Natalie Prange. 2023. A fair and in-depth evaluation of existing end-to-end entity linking systems. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6659–6672, Singapore. Association for Computational Linguistics.

Divya Bhadauria, Alejandro Sierra Múnera, and Ralf Krestel. 2024. The effects of data quality on named entity recognition. In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 79–88. Association for Computational Linguistics.

Yanda Chen, William Gantt, Tongfei Chen, Aaron Steven White, and Benjamin Van Durme. 2023. A unified view of evaluation metrics for structured prediction. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12868–12882, Singapore. Association for Computational Linguistics.

Nancy Chinchor. 1995. MUC-6 Named Entity Task Definition. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Nicola De Cao, Ledell Wu, Kashyap Popat, Mikel Artetxe, Naman Goyal, Mikhail Plekhanov, Luke Zettlemoyer, Nicola Cancedda, Sebastian Riedel, and Fabio Petroni. 2022. Multilingual autoregressive entity linking. *Transactions of the Association for Computational Linguistics*, 10:274–290.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 168–172, Copenhagen, Denmark. Association for Computational Linguistics.

Yifan Ding, Qingkai Zeng, and Tim Weninger. 2024. ChatEL: Entity linking with chatbots. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3086–3097, Torino, Italia. ELRA and ICCL.

Kenneth Enevoldsen, Isaac Chung, Imene Kerboua, Márton Kardos, Ashwin Mathur, David Stap, Jay Gala, Wissam Siblini, Dominik Krzemiński, Genta Winata, Saba Sturua, Saiteja Utpala, Mathieu Ciancone, Marion Schaeffer, Diganta Misra, Shreeya Dhakal, Jonathan Rystrø m, Roman Solomatin, Ömer Çağatan, and 63 others. 2025. Mmteb: Massive multilingual text embedding benchmark. In *International Conference on Learning Representations*, volume 2025, pages 101715–101771.

Ben Hachey, Joel Nothman, and Will Radford. 2014. Cheap and easy entity evaluation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 464–469, Baltimore, Maryland. Association for Computational Linguistics.

Peijie Jiang, Jiapeng Lin, Zhili Wang, Jimeng Sun, and Jiawei Han. 2024. GenRES: Rethinking Evaluation for Generative Relation Extraction in the Era of Large Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2820–2837. Association for Computational Linguistics.

Svetlana Kiritchenko, Stan Matwin, Richard Nock, and Aijun Famili. 2006. Learning and evaluation in the presence of class hierarchies: Application to text categorization. In *Proceedings of the 19th Canadian Conference on Artificial Intelligence*, pages 395–406. Springer.

H. W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

Yangming Li, Lemao Liu, and Shuming Shi. 2021. Empirical analysis of unlabeled entity problem in named entity recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Nut Limsopatham and Nigel Collier. 2016. Bidirectional LSTM for named entity recognition in Twitter messages. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 145–152, Osaka, Japan. The COLING 2016 Organizing Committee.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. Named entity recognition with partially annotated training data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 645—655. Association for Computational Linguistics.

Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. 2023. MTEB: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, Dubrovnik, Croatia. Association for Computational Linguistics.

Isar Nejadgholi, Kathleen C. Fraser, and Berry de Bruijn. 2020. Extensive error analysis and a learning-based evaluation of medical entity recognition systems to approximate user experience. In *Proceedings of the 19th SIGBioMed Workshop on Biomedical Language Processing*, pages 177–186. Association for Computational Linguistics.

Ralph Peeters and Christian Bizer. 2023. Using ChatGPT for entity matching. In *New Trends in Database and Information Systems*, volume 1850 of *Communications in Computer and Information Science*, pages 221–230, Cham. Springer Nature Switzerland.

Ralph Peeters, Aaron Steiner, and Christian Bizer. 2025. Entity matching using large language models. In *Proceedings of the 28th International Conference on Extending Database Technology (EDBT 2025)*, pages 529–541, Barcelona, Spain. OpenProceedings.org.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419. Association for Computational Linguistics.

Maja Popovic, Ekaterina Lapshinova-Koltunski, and Maarit Koponen. 2024. Effects of different types of noise in user-generated reviews on human and machine translations including ChatGPT. In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 17–30. Association for Computational Linguistics.

Frederick Reiss, Hong Xu, Simon Kirsch, and Chengkai Zhu. 2020. Identifying incorrect labels in the CoNLL-2003 corpus. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 215—226. Association for Computational Linguistics.

Andrew Rueda, Elena Alvarez-Mellado, and Constantine Lignos. 2024. CoNLL#: Fine-grained Error

Analysis and a Corrected Test Set for CoNLL-03 English. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3718–3728, Torino, Italia. ELRA and ICCL.

Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. 2013. SemEval-2013 Task 9: Extraction of Drug-Drug Interactions from Biomedical Texts. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147. Association for Computational Linguistics.

Asahi Ushio, Francesco Barbieri, Vitor Sousa, Leonardo Neves, and Jose Camacho-Collados. 2022. Named entity recognition in twitter: A dataset and analysis on short-term temporal shifts. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 309–319. Association for Computational Linguistics.

Asahi Ushio and Jose Camacho-Collados. 2021. T-NER: An all-round python library for transformer-based named entity recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62. Association for Computational Linguistics.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, Daniel Cer, Alice Lisak, Min Choi, Lucas Gonzalez, Omar Sanseviero, Glenn Cameron, Ian Ballantyne, Kat Black, Kaifeng Chen, and 70 others. 2025. Embeddinggemma: Powerful and lightweight text representations. *Preprint*, arXiv:2509.20354.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2025a. GPT-NER: Named entity recognition via large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4257–4275. Association for Computational Linguistics.

Tianshu Wang, Xiaoyang Chen, Hongyu Lin, Xuanang Chen, Xianpei Han, Le Sun, Hao Wang, and Zhenyu Zeng. 2025b. Match, compare, or select? an investigation of large language models for entity matching. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 96–109, Abu Dhabi, UAE. Association for Computational Linguistics.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Griffin Thomas Adams, Jeremy Howard, and Iacopo Poli. 2025. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2526–2547, Vienna, Austria. Association for Computational Linguistics.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, and Michelle Franchini. 2013. OntoNotes release 5.0. Linguistic Data Consortium, LDC2013T19.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 17(2):455–481.

Ikuya Yamada, Hideaki Takeda, and Yoshiyasu Takefuji. 2015. Enhancing named entity recognition in Twitter messages using entity linking. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 136–140, Beijing, China. Association for Computational Linguistics.

Urchade Zaratiana, Gil Pasternak, Oliver Boyd, George Hurn-Maloney, and Ash Lewis. 2025. GLiNER2: Schema-driven multi-task learning for structured information extraction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 130–140, Suzhou, China. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In *Eighth International Conference on Learning Representations (ICLR 2020)*.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Generation and Parsing

We aggregate T-NER output to transform sequences of tokens in spans, GLiNER2 uses its native span output; only GPT 5.4-mini needs a prompt and parser.

Prompt. We query GPT 5.4-mini zero-shot with the template below. `{entity_types}` is the dataset's allowed types as a comma-separated list, and `{text}` the input sentence. We ask the model to produce exact mentions and use the provided types to put it in the best conditions for this task.

```
You are a Named Entity Recognition (NER)
system. Your task is to extract all named
entities from the given text.

Allowed entity types: {entity_types}

Text: "{text}"

Return ONLY a JSON array of objects
matching this schema:
[{"text": "exact entity mention from
the text", "type": "one of the allowed
entity types"}]

Rules:
- If no entities are found, return an
empty array: []
- Do not include explanations, only
output the JSON array.
```

Hyperparameters. The following hyperparameters were used: `top_p` $= 1.0$, `max_tokens` $= 512$, `seed` $= 42$. For GLiNER2, we used `gliner_threshold` $= 0.5$.

Parsing. We parse the response as a JSON array of `{text, type}` objects. Strings are kept verbatim; unparseable responses give an empty prediction set.

B Robustness to Embedding Model

We study how **GeNERscore** and the metrics vary as the embedding model varies. We consider the following models selected from the MTEB/MMTEB leaderboards (Muennighoff et al., 2023; Enevoldsen et al., 2025): Qwen3-Embedding-4B (Zhang et al., 2025) as a representative of big models, Qwen3-Embedding-0.6B (Zhang et al., 2025) as a smaller version of the previous one; EmbeddingGemma-300M (Vera et al., 2025), jina-embeddings-v5-text-small (677M) and jina-embeddings-v5-text-nano (239M) (Akram et al., 2026) as smaller models.

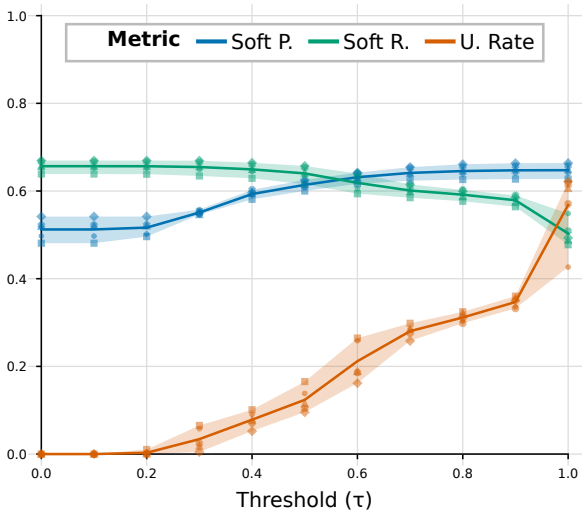


Figure 2: Robustness of the semantic metrics to the embedding model and change of τ/τ_u. Each line is the mean across five encoders and the band their min–max across all datasets and algorithms. For each fixed threshold, the metrics vary little across encoders over the whole τ range. Markers: ■ EmbeddingGemma 300M, ◆ Qwen3-Embedding 0.6B, ▲ Qwen3-Embedding 4B, ✖ Jina v5 Nano, ★ Jina v5 Small.

The choice of encoder has little effect on the metrics. Across five embedding models, *soft precision*, *soft recall*, and the *unsupported rate* stay within a narrow band over the entire τ sweep (Figure 2).

C Assignment Algorithms results

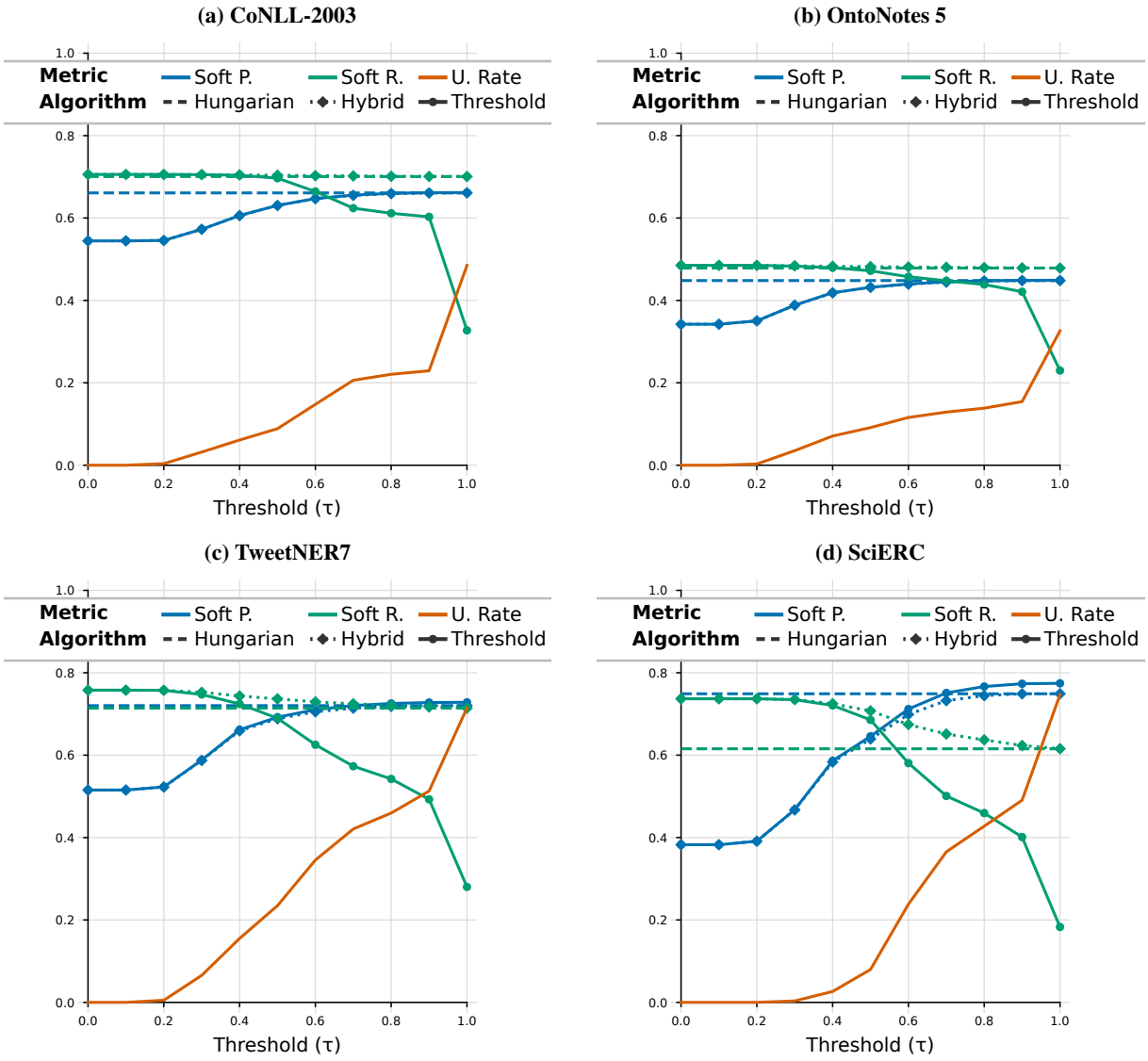


Figure 3: Soft precision (Soft P.), soft recall (Soft R.), and unsupported rate (U. Rate) as a function of $\tau = \tau_u$ for the three assignment algorithms, macro-averaged over T-NER, GLiNER2, and GPT 5.4-mini. Averaged across all embedding models (Appendix B). Hungarian assignment (dashed) is τ-independent. The unsupported rate is assignment-independent and appears as a single orange curve per panel.

D Further Examples

We report 30 further examples from TweetNER7 and SciERC where strict evaluation fails to account for meaning equivalence. For each sample we show gold entities and per-model predictions side by side, with strict F_1, Soft F_1, and unsupported rate U (τ=τ_u=0.6, hybrid assignment, Qwen3-Embedding-0.6B). Each example is tagged with the phenomena it exhibits (§2). Type abbreviations: per = person, loc = location, grp = group, corp = corporation, evt = event, cw = creative_work, prod = product; for SciERC: Meth = Method, Mat = Material, Gen = Generic, Met = Metric, OST = OtherScientificTerm. Characters that cannot be rendered in this document are replaced with descriptive placeholders: [emoji] for emoji, [Thai] for Thai script, and [Cyrillic] for Cyrillic script.

TweetNER7

TweetNER7 #2551 Alias · Partial · Omiss.

"snl had the PERFECT opportunity to do a " target lady during covid " sketch with kristen wiig and chose not too ?!? yeah,I need answers {@Saturday Night Live - SNL@}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
sketch/*prod* target/*corp* {@Saturday Night Live - SNL@}/*prod*	kristen wiig/*per* snl/*grp* {@Saturday Night Live - SNL@}/*prod*	SNL/*corp* kristen wiig/*per* sketch/*cw*	Kristen Wiig/*per* SNL/*corp* Saturday Night Live/*cw* target lady during covid/*cw*
	F_1 .333 $\hat{F}_1$ .434 U .667	F_1 .000 $\hat{F}_1$ .332 U .667	F_1 .000 $\hat{F}_1$ .293 U .750

TweetNER7 #1684 Alias · Paraph. · Partial · Omiss.

"Severe flooding due to Tropical Storm Ulysses Typhoon Vamco Philippines ... {[URL]} via {@YouTube@} # UlyssesPH # PhilippinesNeedsHelp # Philippines # GlobalWarming # Vamco # storm # TropicalCyclone"

Gold	T-NER	GLiNER2	GPT-5.4-mini
Philippines/*loc* Tropical Storm Ulysses/*evt* {@YouTube@}/*prod*	Philippines/*loc* Tropical Storm Ulysses Typhoon Vamco/*evt* {@YouTube@}/*prod*	GlobalWarming/*evt* Philippines/*loc* TropicalCyclone/*evt* Typhoon Vamco/*evt* Vamco/*evt* storm/*evt*	Philippines/*loc* PhilippinesNeedsHelp/*evt* Tropical Storm Ulysses/*evt* TropicalCyclone/*evt* Typhoon Vamco/*evt* UlyssesPH/*evt* Vamco/*evt* YouTube/*corp*
	F_1 .667 $\hat{F}_1$ .972 U .000	F_1 .222 $\hat{F}_1$ .692 U .167	F_1 .364 $\hat{F}_1$ .719 U .250

GPT predicts "Typhoon Vamco" for gold "Tropical Storm Ulysses: the same storm under its international versus Philippine-specific name. A cross-lingual alias that only embedding-based scoring can capture.

TweetNER7 #3 Alias · Type · Omiss.

"- Albin Ekdal is the 1st player from Sweden to score a Serie A goal for {@U.C. Sampdoria@} since Lennart Skoglund on 24 September 1961,also against AC Milan (3-2 away win). # SerieATIM # SampdoriaMilan"

Gold	T-NER	GLiNER2	GPT-5.4-mini
AC Milan/*grp* Albin Ekdal/*per* Lennart Skoglund/*per* Sweden/*loc* {@U.C. Sampdoria@}/*grp*	AC Milan/*grp* Albin Ekdal/*per* Lennart Skoglund/*per* SampdoriaMilan/*grp* Serie A/*evt* Sweden/*loc* {@U.C. Sampdoria@}/*grp*	AC Milan/*corp* Albin Ekdal/*per* Lennart Skoglund/*per* Serie A/*evt* Sweden/*loc*	24 September 1961/*evt* AC Milan/*corp* Albin Ekdal/*per* Lennart Skoglund/*per* SampdoriaMilan/*evt* Serie A/*cw* SerieATIM/*evt* Sweden/*loc* U.C. Sampdoria/*corp*
	F_1 .833 $\hat{F}_1$ .936 U .143	F_1 .600 $\hat{F}_1$ .782 U .200	F_1 .429 $\hat{F}_1$ .768 U .333

All models identify the core entities but disagree on whether sports teams are groups or corporations; the gold also omits in-text mentions such as "Serie A," penalizing models that extract them.

TweetNER7 #43 Type · Omiss.

"I know everyone loves Fauci and govt agencies make mistakes but I really think the CDC needs an overhaul. Needs better leadership,{@Joe Biden@}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
CDC/*corp* {@Joe Biden@}/*per*	CDC/*corp* Fauci/*per* {@Joe Biden@}/*per*	CDC/*corp* Fauci/*corp* govt agencies/*corp*	CDC/*grp* Fauci/*per* Joe Biden/*per*
	F_1 .800 $\hat{F}_1$ .921 U .333	F_1 .400 $\hat{F}_1$ .600 U .000	F_1 .000 $\hat{F}_1$ .752 U .333

TweetNER7 #2364 Alias · Partial · Omiss.

"Everybody is picking the Cowboys,on Gameday shows. Lets prove them wrong. LETS GO BIG BLUE. {@LPG - NYG@} # NYG # NYGvsDAL Andy Dalton talking about watching the Washington game after this game. He thinks they have already won # GiantsPride # Giants"

Gold	T-NER	GLiNER2	GPT-5.4-mini
Cowboys/*grp* Giants/*grp* GiantsPride/*grp* NYG/*grp* NYGvsDAL Andy Dalton/*evt* Washington game/*evt* {@LPG - NYG@}/*per*	Andy Dalton/*per* Cowboys/*grp* Giants/*grp* NYG/*grp* NYGvsDAL/*evt* Washington/*grp* {@LPG - NYG@}/*grp*	Andy Dalton/*per* Cowboys/*grp* Washington/*loc* Washington game/*evt*	Andy Dalton/*per* BIG BLUE/*grp* Cowboys/*grp* Gameday/*cw* Giants/*grp* GiantsPride/*grp* NYG/*grp* NYGvsDAL/*evt* Washington/*grp*
	F_1 .429 $\hat{F}_1$ .805 U .143	F_1 .364 $\hat{F}_1$ .703 U .500	F_1 .500 $\hat{F}_1$ .761 U .222

"BIG BLUE" is a well-known nickname for the New York Giants, an alias that would be penalized even if it was annotated in the gold.

TweetNER7 #2414 Alias · Type · Halluc.

"Jill Biden Leaves the White House in Green Jewel-Toned Jacket & Her Favorite Boot Trend {{URL}} via {@Yahoo@} Hey {@MELANIA TRUMP@},did you see what Dr. Biden was wearing the other day ?"

Gold	T-NER	GLiNER2	GPT-5.4-mini
Boot Trend/*prod* Dr. Biden/*per* Green Jewel-Toned Jacket/*prod* Jill Biden/*per* White House/*loc* {@MELANIA TRUMP@}/*per* {@Yahoo@}/*prod*	Dr. Biden/*per* Jill Biden/*per* White House/*loc* {@MELANIA TRUMP@}/*per* {@Yahoo@}/*prod*	Boot Trend/*prod* Dr. Biden/*per* Green Jewel-Toned Jacket/*prod* Jill Biden/*per* White House/*loc*	Dr. Biden/*per* Jill Biden/*per* MELANIA TRUMP/*per* Whatsup? no/*cw* White House/*loc* Yahoo/*corp*
	F_1 .833 $\hat{F}_1$ .807 U .000	F_1 .833 $\hat{F}_1$ .888 U .000	F_1 .462 $\hat{F}_1$ .648 U .333

GPT hallucinates "Whatsup? no" as an entity while also producing the valid alias "MELANIA TRUMP." "Yahoo" as corporation versus gold product shows type ambiguity. Valid and invalid predictions coexist.

TweetNER7 #307 Alias · Paraph. · Omiss.

"Okay. I 've been a mess all over the weekend because of these award shows. Will watch the replay tom. Congrats,our love {@BTS_twt@} u deserve all of it. I love every single one of you. Yoongi baby,you can finally get some good rest. My 6 tannies,please be safe ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
Yoongi/*per* {@BTS_twt@}/*grp*	Yoongi/*per* {@BTS_twt@}/*grp*	Yoongi baby/*per* award shows/*evt* tom/*per*	BTS_twt/*grp* Yoongi/*per* tannies/*grp*
	F_1 1.00 $\hat{F}_1$ 1.00 U .000	F_1 .000 $\hat{F}_1$ .518 U .333	F_1 .400 $\hat{F}_1$ .845 U .333

GPT predicts "tannies," a fan nickname for BTS that is semantically plausible but absent from gold. Strict matching penalizes the model for capturing world knowledge the annotation misses.

TweetNER7 #782 Type

“Calpulli Mexican Dance Company presents " Dia de los Muertos ," a universal love story that captures the sentiment,color,and hope of day of the dead through music and dance. # DiaDeMuertos # DayoftheDead {{USERNAME}} {{URL}}”

Gold	T-NER	GLiNER2	GPT-5.4-mini
Calpulli Mexican Dance Company/*grp* DayoftheDead/*evt* Dia de los Muertos/*cw* DiaDeMuertos/*evt*	Calpulli Mexican Dance Company/*grp* DayoftheDead/*evt* Dia de los Muertos/*evt* DiaDeMuertos/*evt*	Calpulli Mexican Dance Company/*corp* Dia de los Muertos/*evt* dance/*cw* day of the dead/*evt* music/*cw*	Calpulli Mexican Dance Company/*corp* DayoftheDead/*cw* Dia de los Muertos/*cw*
	F_1 .750 $\hat{F}_1$ .910 U .000	F_1 .000 $\hat{F}_1$ .732 U .400	F_1 .286 $\hat{F}_1$ .675 U .000

TweetNER7 #815 Type · Paraph. · Omiss.

“Watching Tokyo 7th Sister live with {{USERNAME}} and {{USERNAME}} and T7S is lit. Watching Chain The Blossom live,and I really to like to see them live in person soon when I can. I miss eventing in general,but T7S is on my list to see now with friends .”

Gold	T-NER	GLiNER2	GPT-5.4-mini
Chain The Blossom/*evt* Tokyo 7th Sister/*evt*	Chain The Blossom/*grp* T7S/*grp* Tokyo 7th Sister/*grp*	T7S/*corp* T7S/*per* Tokyo 7th Sister live/*evt* eventing/*evt* friends/*grp* {{USERNAME}}/*per*	Chain The Blossom/*cw* T7S/*cw* Tokyo 7th Sister/*cw*
	F_1 .000 $\hat{F}_1$ .648 U .333	F_1 .000 $\hat{F}_1$ .491 U .667	F_1 .000 $\hat{F}_1$ .582 U .333

GPT predicts “T7S,” a standard abbreviation of the gold entity “Tokyo 7th Sister.” Strict matching rejects the alias entirely, soft scoring recovers it.

TweetNER7 #1002 Alias

“" Living is n’t easy but if you can manage to look past your own fears and everyone else ’s expectations,the reward is your life. " Live in the moment. Love every moment. The message in {@Jaden@} Smith ’s latest movie that we could all benefit from. {{URL}}”

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@Jaden@}/*per*	{@Jaden@}/*per*	movie/*cw*	Jaden Smith/*per*
	F_1 1.00 $\hat{F}_1$ 1.00 U .000	F_1 .000 $\hat{F}_1$ .000 U 1.00	F_1 .000 $\hat{F}_1$ .601 U .000

TweetNER7 #1230 Alias · Partial · Halluc.

“# MMATwitter poll tournament Finale {{USERNAME}} Vs {@C H[emoji]T O[emoji]V E R A@} Vs {@Chan Sung Jung@} Vs {@Kevin Holland@} Vote however you like it does n’t have to be who wins the fight it can be who you like more and so forth”

Gold	T-NER	GLiNER2	GPT-5.4-mini
MMATwitter poll tournament Finale/*evt* {@C H[emoji]T O[emoji]V E R A@}/*per* {@Chan Sung Jung@}/*per* {@Kevin Holland@}/*per*	MMATwitter/*evt* tournament Finale/*evt* {@C H[emoji]T O[emoji]V E R A@}/*per* {@Chan Sung Jung@}/*per* {@Kevin Holland@}/*per*	USERNAME/*per* fight/*evt* tournament Finale/*evt*	Chan Sung Jung/*per* Kevin Holland/*per* MMATwitter/*grp* Whatsup/*per*
	F_1 .667 $\hat{F}_1$ .852 U .000	F_1 .000 $\hat{F}_1$ .249 U .667	F_1 .000 $\hat{F}_1$ .622 U .250

A mix of valid aliases (“Chan Sung Jung,” “Kevin Holland” stripped of markup) and a genuine hallucination (“Whatsup” as a person). Soft scoring rewards the former while the unsupported rate flags the latter.

TweetNER7 #1266 Alias

“i have a really good feeling the trailer is coming out in like 5 minutes since it ’s gonna be 12 please disney {@Disney TVA@}”

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@Disney TVA@}/*corp*	{@Disney TVA@}/*prod*	disney/*corp* trailer/*cw*	Disney TVA/*corp* disney/*corp*
	F_1 .000 $\hat{F}_1$ .654 U .000	F_1 .000 $\hat{F}_1$ .683 U .500	F_1 .000 $\hat{F}_1$ .856 U .000

TweetNER7 #1284 Alias

"Please please please,next week re shuffle the guys. . too boring # TV3GH # DateRush {@G i o v a n i CALEB #DATErush@}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@G i o v a n i CALEB #DATErush@}/*per*	{@G i o v a n i CALEB #DATErush@}/*per*	CALEB/*per* DateRush/*evt* TV3GH/*prod* guys/*grp*	DATErush/*cw* DateRush/*cw* Giovani Caleb/*per* TV3GH/*corp*
	F_1 1.00 $\hat{F}_1$ 1.00 U .000	F_1 .000 $\hat{F}_1$ .570 U .750	F_1 .000 $\hat{F}_1$ .570 U .750

TweetNER7 #1350 Alias · Type · Partial · Omiss.

"Excited to be on the beat for tonight 's Rangers-Devils tilt The first of four games between this team this week from the Rock begins at 7 Nico Hischier returns to the lineup for NJ {{USERNAME}} # NYR # NJDevils"

Gold	T-NER	GLiNER2	GPT-5.4-mini
NJDevils/*cw* Nico/*loc* Rangers-Devils/*evt* games/*grp*	NJ/*grp* NJDevils/*grp* NYR/*grp* Nico Hischier/*per* Rangers-Devils tilt/*evt*	NJ/*corp* Nico Hischier/*per*	Devils/*grp* NJ/*loc* NJDevils/*grp* NYR/*grp* Nico Hischier/*per* Rangers/*grp* the Rock/*loc*
	F_1 .000 $\hat{F}_1$ .602 U .400	F_1 .000 $\hat{F}_1$ .000 U 1.00	F_1 .000 $\hat{F}_1$ .646 U .143

"Devils" is a valid short form of "NJDevils," and "NJDevils" is typed as "group" versus gold "creative_work.".

TweetNER7 #1467 Paraph. · Partial

"< strong >26 Ways I Can Do More to Support the Black Lives Matter Movement </ strong >{{URL}} via {{USERNAME}}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
Black Lives Matter/*grp*	Black Lives Matter Movement/*grp*	Black Lives Matter Movement/*grp* USERNAME/*per*	26 Ways I Can Do More to Support the Black Lives Matter Movement/*cw* Black Lives Matter Movement/*grp*
	F_1 .000 $\hat{F}_1$ .967 U .000	F_1 .000 $\hat{F}_1$ .775 U .500	F_1 .000 $\hat{F}_1$ .840 U .500

"Black Lives Matter Movement" is an expansion of the gold entity "Black Lives Matter." Soft scoring correctly rewards this near-match that strict matching rejects.

TweetNER7 #1969 Omiss.

"Hi ! I 'm originally an iKONIC and a teume since ygtb era and it 's just sad that i do n't any teume / iKONIC moots It 's my first stan account so please help me gain some moots. Pls rt & ff me,I will definitely ff you back {{USERNAME}}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
iKONIC/*per*	iKONIC/*grp*	USERNAME/*per* iKONIC/*prod* moots/*prod* stan account/*prod* teume/*prod*	iKONIC/*grp* teume/*grp* ygtb/*cw*
	F_1 .000 $\hat{F}_1$ .612 U .000	F_1 .000 $\hat{F}_1$ .546 U .600	F_1 .000 $\hat{F}_1$ .530 U .667

Gold labels "iKONIC" as a person, but it is a K-pop fan group name; models labeling it as "group" are arguably more correct than the gold annotation.

TweetNER7 #2195 Omiss.

"Random It is predicted that the reason why night insects,such as moths,are attracted to lights is because they mistake them for the light of the moon,which they used to navigate the Earth before mankind made artificial lights. {@MISSIO@} # CanYouFeelTheSun {{USERNAME}} # alt18"

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@MISSIO@}/*grp*	{@MISSIO@}/*per*	Earth/*loc* artificial lights/*prod* lights/*prod* mankind/*grp*	CanYouFeelTheSun/*cw* Earth/*loc*
	F_1 .000 $\hat{F}_1$ .612 U .000	F_1 .000 $\hat{F}_1$ .000 U 1.00	F_1 .000 $\hat{F}_1$ .000 U 1.00

TweetNER7 #2250 Alias · Omiss.

"HONESTLY I'M VERY CONCERNED PLEASE VOTE FOR TSA !!! THIS IS ONE WE ABSOLUTELY CANNOT LOSE # BTS {@BTS_twt@} {{URL}}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
TSA/*grp* {@BTS_twt@}/*grp*	TSA/*grp* {@BTS_twt@}/*grp*	TSA/*corp*	BTS/*grp* TSA/*grp*
	F_1 1.00 $\hat{F}_1$ 1.00 U .000	F_1 .000 $\hat{F}_1$ .471 U .000	F_1 .500 $\hat{F}_1$ .616 U .000

GPT predicts "BTS" for the gold handle "{@BTS_twt@}":the same referent under a different surface form.

TweetNER7 #2428 Alias · Omiss.

"' THE OCEAN EYES ' I love the sound of this. {@mewsuppasit@} # MewSuppasit #[Thai]"

Gold	T-NER	GLiNER2	GPT-5.4-mini
MewSuppasit/*grp* THE OCEAN EYES/*cw* {@mewsuppasit@}/*per*	MewSuppasit/*per* THE OCEAN EYES/*cw* {@mewsuppasit@}/*per*	THE OCEAN EYES/*cw*	MewSuppasit/*per* THE OCEAN EYES/*cw* [Thai]/*per*
	F_1 .667 $\hat{F}_1$ .885 U .000	F_1 .500 $\hat{F}_1$ .500 U .000	F_1 .333 $\hat{F}_1$ .779 U .333

GPT predicts the Thai name [Thai] of the same person as the gold handle alias. Gold also labels "MewSuppasit" as a group rather than a person.

TweetNER7 #2547 Alias · Omiss.

"I hope you live in a place where the sky is blue enough,the sunshine is good enough,the shops on the street are clean and delicious,and the lights are always on at home. {@mewsuppasit@} # MewSuppasit #[Thai]"

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@mewsuppasit@}/*per*	MewSuppasit/*per* {@mewsuppasit@}/*per*	home/*loc* lights/*prod* shops/*loc* sky/*loc* street/*loc* sunshine/*loc*	MewSuppasit/*per* mewsuppasit/*per* [Thai]/*per*
	F_1 .667 $\hat{F}_1$ .946 U .000	F_1 .000 $\hat{F}_1$ .000 U 1.00	F_1 .000 $\hat{F}_1$ .812 U .333

Same pattern as example TweetNER7#2428 above

TweetNER7 #2631 Alias · Partial

"The developer and landlord class will not cease until Dublin City is nothing more than a series of hotels An Bord Pleanála : Save James Joyce 's ' House of the Dead ' (Usher 's Island) from becoming a tourist hostel. - Sign the Petition ! {{URL}} via {@Change org [Cyrillic]@}"

Gold	T-NER	GLiNER2	GPT-5.4-mini
Dublin City/*loc* House of the Dead/*cw* Island/*loc* James Joyce/*per* Usher/*loc* developer and landlord class/*grp* {@Change org [Cyrillic]@}/*prod*	An Bord Pleanála/*grp* Dublin City/*loc* House of the Dead/*cw* James Joyce/*per* Usher 's Island/*loc* {@Change org [Cyrillic]@}/*prod*	Dublin City/*loc* House of the Dead/*cw* James Joyce/*per* developer/*corp*	An Bord Pleanála/*corp* Change org Russia/*corp* Dublin City/*loc* House of the Dead/*cw* James Joyce/*per* Usher's Island/*loc*
	F_1 .615 $\hat{F}_1$ .838 U .167	F_1 .545 $\hat{F}_1$ .571 U .250	F_1 .462 $\hat{F}_1$ .719 U .333

GPT predicts "Change org Russia." which is an alias for "{@Change org [Cyrillic]@}"

TweetNER7 #2711 Alias · Partial

"Work hard,have fun,make history {{USERNAME}} WE LOVE YOU SOHEL # HBDIsmartSohel # HBDSyedSohelRyan"

Gold	T-NER	GLiNER2	GPT-5.4-mini
HBDIsmartSohel/*per* HBDSyedSohelRyan/*per*	HBDIsmartSohel/*grp* HBDSyedSohelRyan/*grp* SOHEL/*per*	SOHEL/*per*	Sohel/*per* Syed Sohel Ryan/*per*
	F_1 .000 $\hat{F}_1$ .716 U .000	F_1 .000 $\hat{F}_1$ .775 U .000	F_1 .000 $\hat{F}_1$ .847 U .000

GPT extracts "Sohel" and "Syed Sohel Ryan" from birthday hashtags: correct referents that strict matching rejects because the surface forms differ from the hashtag gold.

TweetNER7 #2776 Alias · Omiss.

"Flawless performance by Izzy {@Israel Adesanya@}. {@Paulo Costa (Borrachinha)@} looked defeated before the fight begun. To his corner - There 's no point in hyping your fighter hoping he wins and you get a piece of that cake. Confidence comes with experience ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@Israel Adesanya@}/*per* {@Paulo Costa (Borrachinha)@}/*per*	fight/*evt* {@Israel Adesanya@}/*per* {@Paulo Costa (Borrachinha)@}/*per*	Borrachinha/*per* Izzy/*per* cake/*prod* corner/*loc* fight/*evt* fighter/*per*	Borrachinha/*per* Israel Adesanya/*per* Izzy/*per* Paulo Costa/*per*
	F_1 .800 $\hat{F}_1$ .874 U .333	F_1 .000 $\hat{F}_1$ .622 U .667	F_1 .000 $\hat{F}_1$ .826 U .000

"Izzy" is a nickname for gold "Israel Adesanya," and "Borrachinha"/"Paulo Costa" are alternate forms of the compound gold name. Only semantic scoring credits these valid aliases.

TweetNER7 #2793 Alias · Type

"I have spent a dumb amount of time mapping out my first water cooling loop. I keep modifying it every few days. Really need to make a decision before {@Cyberpunk 2077@} comes out ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
{@Cyberpunk 2077@}/*prod*	{@Cyberpunk 2077@}/*cw*	water cooling loop/*prod*	Cyberpunk 2077/*cw*
	F_1 .000 $\hat{F}_1$ .714 U .000	F_1 .000 $\hat{F}_1$ .000 U 1.00	F_1 .000 $\hat{F}_1$ .650 U .000

SciERC

SciERC #47 Type · Partial · Omiss.

"Towards deep analysis of compositional classes of paraphrases,we have examined a class-oriented framework for collecting paraphrase examples,in which sentential paraphrases are collected for each paraphrase class separately by means of automatic candidate generation and manual judgement. Our preliminary experiments on building a paraphrase corpus have so far been producing promising results,which we have evaluated according to cost-efficiency,exhaustiveness,and reliability ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
automatic candidate generation/*Meth* class-oriented framework/*Meth* compositional classes of paraphrases/*Task* manual judgement/*Meth* paraphrase corpus/*Mat* paraphrase examples/*Mat* sentential paraphrases/*Mat*	automatic candidate generation/*Meth* class-oriented framework/*Meth* compositional classes of paraphrases/*OST* deep analysis of/*Task* manual judgement/*OST* paraphrase/*OST* paraphrase corpus/*OST* paraphrase examples/*OST* sentential paraphrases/*OST*	automatic candidate generation/*Meth* cost-efficiency/*Met* exhaustiveness/*Met* experiments/*Task* manual judgement/*Meth* paraphrase corpus/*Gen* paraphrases/*Gen* reliability/*Met* sentential paraphrases/*OST*	automatic candidate generation/*Meth* class-oriented framework/*Meth* compositional classes of paraphrases/*OST* cost-efficiency/*Met* deep analysis/*Task* exhaustiveness/*Met* manual judgement/*Meth* paraphrase class/*OST* paraphrase corpus/*OST* paraphrase examples/*OST* reliability/*Met* sentential paraphrases/*OST*
	F_1 .250 $\hat{F}_1$ .477 U .778	F_1 .250 $\hat{F}_1$ .382 U .778	F_1 .316 $\hat{F}_1$ .500 U .750

All models struggle with SciERC's fine-grained type distinctions (Task versus OtherScientificTerm).

SciERC #56 Type · Partial · Omiss.

"We investigate the problem of learning to predict moves in the board game of Go from game records of expert players. In particular,we obtain a probability distribution over legal moves for professional play in a given position. This distribution has numerous applications in computer Go,including serving as an efficient stand-alone Go player. It would also be effective as a move selector and move sorter for game tree search and as a training tool for Go players. Our method has two major components : a) a pattern extraction scheme for efficiently harvesting patterns of given size and shape from expert game records and b) a Bayesian learning algorithm (in two variants) that learns a distribution over the values of a move given a board position based on the local pattern context. The system is trained on 181,000 expert games and shows excellent prediction performance as indicated by its ability to perfectly predict the moves made by professional Go players in 34 % of test positions ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
Bayesian learning algorithm/*Meth* Go players/*OST* It/*Gen* board game of Go/*Task* computer Go/*OST* distribution/*Gen* expert games/*Mat* game records of expert players/*Mat* game tree search/*Meth* local pattern context/*OST* method/*Gen* move selector/*Meth* move sorter/*Meth* pattern extraction scheme/*Meth* probability distribution/*OST* stand-alone Go player/*OST* system/*Gen* training tool/*Task*	Bayesian learning algorithm/*Meth* Go/*Meth* It/*Gen* board game/*Task* computer Go/*Task* distribution/*Gen* game tree search/*Task* method/*Gen* move selector/*Meth* move sorter/*Meth* pattern extraction scheme/*Meth* predict/*Task* probability distribution/*OST* stand-alone/*Meth* training/*Task*	Bayesian learning algorithm/*Meth* Go/*Gen* LRB/*Meth* RRB/*Meth* computer Go/*Gen* expert players/*Gen* game records/*Mat* game tree search/*Task* legal moves/*OST* local pattern context/*OST* move selector/*Task* move sorter/*Task* pattern extraction scheme/*Meth* prediction performance/*Met* probability distribution/*OST* professional play/*Task* training tool/*Task*	181,000 expert games/*Met* 34 %/*Met* Bayesian learning algorithm/*Meth* Go/*Gen* Go player/*Gen* board position/*Gen* computer Go/*Gen* distribution/*OST* excellent prediction performance/*Met* expert players/*Gen* game records/*OST* game tree search/*Task* legal moves/*Task* local pattern context/*OST* move selector/*Meth* move sorter/*Meth* pattern extraction scheme/*Meth* patterns/*OST* test positions/*Met* training tool/*Meth*
	F_1 .485 $\hat{F}_1$ .693 U .133	F_1 .286 $\hat{F}_1$ .691 U .353	F_1 .263 $\hat{F}_1$ .648 U .350

Dense entity extraction with many partial overlaps ("Go" versus "Go players," "game records" versus "game records of expert

players"). Soft scoring captures proportional credit that strict matching denies.

SciERC #64 Alias · Type · Partial · Omiss.

"This work proposes a new research direction to address the lack of structures in traditional n-gram models. It is based on a weakly supervised dependency parser that can model speech syntax without relying on any annotated training corpus. Labeled data is replaced by a few hand-crafted rules that encode basic syntactic knowledge. Bayesian inference then samples the rules,disambiguating and combining them to create complex tree structures that maximize a discriminative model 's posterior on a target unlabeled corpus. This posterior encodes sparse se-lectional preferences between a head word and its dependents. The model is evaluated on English and Czech newspaper texts,and is then validated on French broadcast news transcriptions ."

Gold	**T-NER**	**GLiNER2**	**GPT-5.4-mini**
Bayesian inference/*Meth*	Bayesian inference/*Meth*	Bayesian inference/*Meth*	Bayesian inference/*Meth*
English and Czech newspaper texts/*Mat*	English and Czech newspaper texts/*Mat*	Czech/*Gen*	Czech/*Gen*
French broadcast news transcriptions/*Mat*	It/*Gen*	English/*Gen*	English/*Gen*
Labeled data/*OST*	Labeled data/*OST*	dependency parser/*Meth*	French/*Gen*
annotated training corpus/*Mat*	annotated training corpus/*Mat*	dependents/*OST*	annotated training corpus/*Gen*
complex tree structures/*OST*	complex tree structures/*OST*	head word/*OST*	broadcast news transcriptions/*Gen*
discriminative model 's posterior/*OST*	discriminative model/*Meth*	n-gram models/*OST*	dependents/*Gen*
hand-crafted rules/*OST*	hand-crafted rules/*OST*	posterior/*Met*	discriminative model/*Meth*
lack of structures/*OST*	head/*OST*	se-lectional preferences/*OST*	hand-crafted rules/*Meth*
lack of structures in traditional n-gram models/*Task*	model/*Gen*	speech syntax/*OST*	head word/*OST*
model/*Gen*	n-gram models/*Meth*	syntactic knowledge/*OST*	n-gram models/*OST*
n-gram models/*Meth*	posterior/*OST*	training corpus/*Mat*	newspaper texts/*Gen*
posterior/*Gen*	sparse se-lectional preferences/*OST*	tree structures/*OST*	posterior/*Met*
rules/*OST*	speech syntax/*OST*		rules/*Gen*
sparse se-lectional preferences/*OST*	syntactic knowledge/*OST*		selectional preferences/*OST*
speech syntax/*OST*	unlabeled corpus/*Mat*		speech syntax/*OST*
syntactic knowledge/*OST*	weakly supervised dependency parser/*Meth*		syntactic knowledge/*OST*
them/*Gen*			target unlabeled corpus/*Gen*
unlabeled corpus/*Mat*			tree structures/*OST*
weakly supervised dependency parser/*Task*			weakly supervised dependency parser/*Meth*
	F_1 .649 $\hat{F}_1$ .839 U .000	F_1 .182 $\hat{F}_1$ .577 U .154	F_1 .150 $\hat{F}_1$ .581 U .150

SciERC #69 Type

"The goal of this work is the enrichment of human-machine interactions in a natural language environment. Because a speaker and listener can not be assured to have the same beliefs,contexts,perceptions,backgrounds,or goals,at each point in a conversation,difficulties and mistakes arise when a listener interprets a speaker 's utterance. These mistakes can lead to various kinds of misunderstandings between speaker and listener,including reference failures or failure to understand the speaker 's intention. We call these misunderstandings miscommunication. Such mistakes can slow,and possibly break down,communication. Our goal is to recognize and isolate such miscommunications and circumvent them. This paper highlights a particular class of miscommunication – reference problems – by describing a case study and techniques for avoiding failures of reference. We want to illustrate a framework less restrictive than earlier ones by allowing a speaker leeway in forming an utterance about a task and in determining the conversational vehicle to deliver it. The paper also promotes a new view for extensional reference ."

Gold	**T-NER**	**GLiNER2**	**GPT-5.4-mini**
extensional reference/*Task* failures of reference/*Task* human-machine interactions/*Task* miscommunication/*Task* miscommunications/*Task* natural language environment/*OST* reference failures/*OST* reference problems/*Task* speaker 's intention/*OST* techniques/*Gen* them/*Gen*	human-machine/*Task* miscommunication/*OST* miscommunications/*OST* misunderstandings/*OST* natural language/*OST* reference/*OST*	backgrounds/*OST* beliefs/*OST* case study/*Meth* contexts/*OST* extensional reference/*OST* framework/*OST* goals/*OST* human-machine interactions/*OST* miscommunication/*OST* misunderstandings/*OST* perceptions/*OST* reference failures/*OST* reference problems/*OST* task/*Task* techniques/*Meth* utterance/*Task*	case study/*Meth* extensional reference/*OST* human-machine interactions/*Task* miscommunication/*OST* natural language environment/*OST* reference problems/*Task*
	F_1 .000 $\hat{F}_1$ .403 U .000	F_1 .074 $\hat{F}_1$ .571 U .250	F_1 .353 $\hat{F}_1$ .608 U .167

T-NER scores zero strict F_1 despite predicting most gold entities, solely because type mismatches (Task versus OtherScientificTerm). Soft scoring recovers to ≈ 0.4. GPT and GLiNER2 retrieve better results.

SciERC #66 Alias · Type · Partial

"We present a novel method for discovering parallel sentences in comparable,non-parallel corpora. We train a maximum entropy classifier that,given a pair of sentences,can reliably determine whether or not they are translations of each other. Using this approach,we extract parallel data from large Chinese,Arabic,and English non-parallel newspaper corpora. We evaluate the quality of the extracted data by showing that it improves the performance of a state-of-the-art statistical machine translation system. We also show that a good-quality MT system can be built from scratch by starting with a very small parallel corpus (100,000 words) and exploiting a large non-parallel corpus. Thus,our method can be applied with great benefit to language pairs for which only scarce resources are available ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
Chinese,Arabic,and English non-parallel newspaper corpora/*Mat* MT system/*Meth* approach/*Gen* comparable,non-parallel corpora/*Mat* discovering parallel sentences/*Task* it/*Gen* maximum entropy classifier/*Meth* method/*Gen* non-parallel corpus/*Mat* parallel corpus/*Mat* parallel data/*Mat* scarce resources/*Mat* statistical machine translation system/*Meth*	Chinese,Arabic,and English non-parallel newspaper corpora/*Mat* MT system/*Meth* approach/*Gen* comparable,non-parallel corpora/*Mat* it/*Gen* maximum entropy classifier/*Meth* method/*Gen* parallel sentences/*Mat* statistical machine translation system/*Meth*	Arabic/*Gen* Chinese/*Gen* English/*Gen* MT system/*Meth* language pairs/*Gen* maximum entropy classifier/*Meth* performance/*Met* statistical machine translation system/*Meth* translations/*Task*	100,000 words/*Met* Arabic/*Gen* Chinese/*Gen* English/*Gen* MT system/*Task* comparable/*OST* language pairs/*Gen* maximum entropy classifier/*Meth* method/*Meth* non-parallel corpora/*Task* non-parallel corpus/*Task* non-parallel newspaper corpora/*Task* pair of sentences/*Gen* parallel corpus/*Task* parallel data/*Task* parallel sentences/*Task* quality/*Met* scarce resources/*OST* state-of-the-art statistical machine translation system/*Task* this approach/*Meth* translations/*OST*
	F_1 .727 $\hat{F}_1$ .809 U .000	F_1 .273 $\hat{F}_1$ .441 U .222	F_1 .059 $\hat{F}_1$ .547 U .238

SciERC #85 Alias · Type · Paraph. · Omiss.

"Sentence boundary detection in speech is important for enriching speech recognition output,making it easier for humans to read and downstream modules to process. In previous work,we have developed hidden Markov model (HMM) and maximum entropy (Maxent) classifiers that integrate textual and prosodic knowledge sources for detecting sentence boundaries. In this paper,we evaluate the use of a conditional random field (CRF) for this task and relate results with this model to our prior work. We evaluate across two corpora (conversational telephone speech and broadcast news speech) on both human transcriptions and speech recognition output. In general,our CRF model yields a lower error rate than the HMM and Max-ent models on the NIST sentence boundary detection task in speech,although it is interesting to note that the best results are achieved by three-way voting among the classifiers. This probably occurs because each model has different strengths and weaknesses for modeling the knowledge sources ."

Gold	T-NER	GLiNER2	GPT-5.4-mini
CRF model/*Meth* HMM and Max-ent models/*Meth* NIST sentence boundary detection task/*Mat* Sentence boundary detection/*Task* broadcast news speech/*Mat* classifiers/*Meth* conditional random field (CRF)/*Meth* conversational telephone speech/*Mat* corpora/*Gen* detecting sentence boundaries/*Task* error rate/*Met* hidden Markov model (HMM) and maximum entropy (Maxent) classifiers/*Meth* human transcriptions/*OST* it/*Gen* knowledge sources/*Mat* model/*Gen* speech/*Mat* speech recognition output/*OST* task/*Gen* textual and prosodic knowledge sources/*Mat* three-way voting/*Meth*	Sentence boundary detection in/*Task* conditional random field (CRF)/*Meth* conversational telephone speech/*Mat* detecting/*Task* hidden Markov model (HMM)/*Meth* maximum entropy (Maxent) classifiers/*Meth* model/*Gen* sentence boundaries/*OST* speech/*Mat* speech recognition output/*Task* task/*Gen* textual and prosodic knowledge sources/*OST*	Sentence boundary detection/*Task* conditional random field/*Meth* corpora/*Mat* error rate/*Met* human transcriptions/*Mat* humans/*Gen* knowledge sources/*OST* speech/*Gen* three-way voting/*Meth*	CRF/*Meth* HMM/*Meth* Max-ent/*Meth* Maxent/*Meth* NIST sentence boundary detection task in speech/*Task* Sentence boundary detection in speech/*Task* broadcast news speech/*Gen* classifiers/*Meth* conditional random field/*Meth* conversational telephone speech/*Gen* error rate/*Met* hidden Markov model/*Meth* human transcriptions/*Gen* knowledge sources/*OST* maximum entropy/*Meth* speech recognition output/*Task* three-way voting/*Meth*
	F_1 .303 $\hat{F}_1$ .708 U .083	F_1 .200 $\hat{F}_1$ .705 U .000	F_1 .158 $\hat{F}_1$ .595 U .059

All models extract abbreviation aliases ("CRF" for "CRF model," "HMM" for the full name) that strict matching rejects. Soft scoring recovers substantial credit across all three.

How Noisy is Automatic Annotation? Characterizing Label Noise from Generative Models in Multi-Dimensional Spanish Subjectivity Annotation

Alba Pérez-Montero,[1] **Elena Lloret,**[2] **Paloma Moreda**[2]
[1]University Institute for Computing Research (IUII), University of Alicante
[2]Dept. of Software and Computing Systems, University of Alicante
alba.perezm@ua.es, {elloret,moreda}@dlsi.ua.es

Abstract

Although human variability has traditionally been the primary source of annotation noise, the increasing use of Large Language Models (LLMs) for automated labeling is shifting this landscape toward systematic, machine-generated label noise. This challenge is particularly severe in Subjectivity Detection (SD), where the behavior of automated errors remains largely unexamined. This study characterizes the extent and linguistic nature of automated annotation noise in Spanish SD. Errors are evaluated as omissions (under-generation) or additions (over-generation) across six subjectivity dimensions: Doubt, Certainty, Specificity, Non-specificity, Distancing, and Participation. Using an expert-annotated human gold standard as a reference, a traditional machine learning control is benchmarked against three state-of-the-art LLM families (GPT, Gemini, and Llama). The results reveal a clear architectural divide: while the traditional ML framework predominantly generates omission errors (63.99%), only detecting markers it was explicitly programmed to recognize, LLMs like Gemini and Llama lean heavily toward addition errors (exceeding 51%), guessing and hallucinating markers instead of leaving uncertain cases unlabeled. Although Gemini minimizes overall noise, achieving F1-scores up to 0.96 in specific dimensions, Certainty remains highly susceptible to errors, showing a 53.96% error rate in the traditional ML control. Ultimately, this diagnostic evaluation of LLMs as annotators in this specific subjective task reveals that automated systems introduce systematic label noise, demonstrating that human-in-the-loop participation is indispensable to resolve these errors and ensure reliable dataset curation.

1 Introduction

The boundary between natural user-generated text (UGT) and AI-generated content is becoming increasingly blurred. Instead of just analyzing digital discourse, Large Language Models (LLMs) are now widely used to automatically annotate datasets (Klie et al., 2023). Consequently, data corruption in modern NLP pipelines has transitioned from random human variance into systematic, AI-induced generative label noise (Nasution and Onan, 2024). When LLMs attempt to interpret subjective, informal, or non-standard online text, they frequently introduce this noise.

From a noisy-text perspective, these annotation errors are more than just low performance metrics; they represent a distinct form of LLM-generated noisy text that can compromise both dataset quality and the training of downstream models. This issue is especially clear when annotating natural online discourse, which is highly subjective. Pragmatic and discursive tasks, such as Subjectivity Detection (SD), are widely recognized as some of the most complex annotation challenges, being highly susceptible to errors (Wiebe et al., 2005). Because expressing subjectivity is a highly individual act (Bajtin, 1982), capturing its nuances requires interpreting the speaker's stance and intent within a specific context (Wilson et al., 2005). When annotating these subtle dimensions, LLMs often struggle with linguistic ambiguity, resulting in systematic errors, particularly in non-English languages like Spanish. Despite the risks of automatic annotation, how this LLM-generated noise behaves and distributes across different subjectivity dimensions remains largely unexamined.

To address this gap, this study presents an exploratory diagnosis of LLM-generated label noise in Spanish subjectivity annotation. As previously shown by Piot et al. (2026), traditional evaluation metrics fall short when assessing LLMs as reliable annotators in subjective tasks. For this reason, instead of relying on broad statistical averages, a granular noise-codification framework is proposed to analyze under-generation (*omissions*) and over-generation (*additions*) in high-density, carefully an-

Proceedings of the 11th Workshop on Natural User-generated Text, pages 141–153
October 28, 2026 ©2026 Association for Computational Linguistics

notated texts. This systematic approach pursues a three-fold objective: i) to assess the viability of automated annotation in these high-density settings; ii) to deploy the proposed diagnostic framework to characterize model-specific error profiles; and iii) to analyze how generative models handle label noise in pragmatically complex discourse.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 introduces the multi-dimensional subjectivity annotation scheme and the proposed noise diagnostic methodology. The experimental setup, including the evaluation dataset and the comparison between traditional and LLM pipelines, is detailed in Section 5. Section 6 discusses the quantitative and qualitative findings regarding AI-induced label noise, and Section 7 concludes the paper and outlines its limitations.

2 Related Work

Historically, text noisiness from a data quality perspective focused primarily on surface-level defects, such as typographical errors and OCR corruption (Bhadauria et al., 2024). However, as modern language models have achieved broad proficiency in general linguistic tasks (Chang et al., 2024), the research focus has shifted toward deeper, more complex tiers of data corruption driven by their deployment as automated, silver-standard annotators (Jahan et al., 2024). Within this paradigm, the primary bottleneck is no longer surface-level accuracy. Instead, the challenge shifts to pragmatic and discursive layers, where the models' limitations in metalinguistic analysis and complex reasoning lead to systematic errors (Amin et al., 2024).

When LLMs are used for automated annotation, this limitation often results in a clear trade-off, yielding high recall but lower precision (Nasution and Onan, 2024). This vulnerability becomes even more severe when handling highly ambiguous tasks (Voutsa et al., 2025). Even when evaluated across sophisticated linguistic benchmarks (Srivastava et al., 2023; Ettinger et al., 2023), LLMs frequently produce human-like outputs that lack true metalinguistic comprehension (Song et al., 2025), which reveals that models can offer a comparative evaluation path, despite presenting instance-level divergence in complex annotation tasks (Piot et al., 2026). Recent work highlight that traditional evaluation approaches on LLMs as annotators for complex tasks, such as hate speech or cognitive distortions detection, frequently fall short because the variability performance under subjective conditions is not considered (Piot et al., 2026; Sharma et al., 2026; Xu et al., 2026).

Regarding LLMs as annotators, the lack of metalinguistic comprehension is highly visible in subjectivity analysis, which focuses on evaluating authorial involvement and stance within a message (Kasmuri and Basiron, 2017). At a conceptual level, subjectivity represents the explicit or implicit commitment of the author, carrying rich pragmatic features, including personal opinions and intentions, that depend heavily on context (Hyland, 2005). Because capturing these nuances requires deep inferential reasoning, automated annotation methods struggle to maintain data validity in this domain (Shokri et al., 2024). Evaluating LLMs is therefore necessary to determine if generative architectures can handle such non-surface-level linguistic tasks. Conversely, previous work showed that LLMs can serve as consistent annotators for inherently subjective tasks by reframing the annotation process as a repeated querying rather than a single shot prediction (Sharma et al., 2026).

Besides, this shift toward evaluating generative models contrasts with traditional dataset curation, where Annotation Error Detection (AED) has historically focused on identifying human inconsistencies using structural or lexical patterns (Dickinson and Meurers, 2003; Klie et al., 2023). However, raw quantitative error metrics often mask qualitative failures in nuanced linguistic domains (Barnes et al., 2019). This limitation highlights the need for more granular evaluation methods to ensure dataset quality and scientific validity (Dickinson, 2015).

This generative noise becomes especially prominent in pragmatic tasks where models struggle to interpret implicit markers, causing them to diverge from human gold standards (Schroeder et al., 2025). Crucially, the current literature lacks diagnostic frameworks to characterize the qualitative profiles of the noise introduced by automated annotations (Klie et al., 2023). This study addresses this gap by proposing an exploratory evaluation framework designed to profile and analyze AI-induced label noise in Spanish subjectivity tagging, shifting the analytical focus from simple error detection to granular noise characterization.

3 Multi-Dimensional Subjective Detection Annotation

Grounded in Hyland (2005) and Martín Zorraquino (1999), the annotation scheme employed in this study is built around the notion of *authorial commitment*: the degree to which a speaker presents a proposition as certain, concrete, and personally endorsed, rather than tentative or detached. Linguistic devices that weaken this commitment are known as *hedges* (e.g., the Spanish subjunctive, or adverbs like *"maybe"*), while devices that strengthen it are known as *boosters* (e.g., the imperative mood, or adverbs like *"yes"*). This distinction maps directly onto three conceptual axes of pragmatic variance: certainty, concreteness, and involvement. Rather than isolated categories, each axis functions as a bipolar continuum pairing a low-commitment hedge (LC) against a high-commitment booster (HC) counterpart. This balanced structure yields six distinct dimensions:

- **Degree of Certainty (Doubt vs. Certainty):** Evaluates authorial commitment to truth by contrasting hedging markers, such as the subjunctive mood (*Doubt (LC)*), with boosting indicators of conviction (*Certainty (HC)*).
- **Level of Concreteness (Non-specificity vs. Specificity):** Distinguishes between vague, generalizing terms that hedge accountability (*Non-specificity (LC)*) and boosting referential anchors like named entities (*Specificity (HC)*).
- **Scale of Involvement (Distancing vs. Participation):** Measures the author's discursive presence by contrasting hedging, impersonal or passive constructions (*Distancing (LC)*) with boosting, direct engagement markers (*Participation (HC)*).

Although traditional pragmatic analysis is typically context-based, this scheme is applied at the word or span level, following previous approaches to subjectivity analysis (Vieira et al., 2020; Antici et al., 2021) that have demonstrated the benefits of this granularity. It is designed as a multi-label framework where annotations target specific linguistic markers that are the concrete materialization of the subjectivity dimensions described above. Because a single word or phrase can perform multiple pragmatic functions simultaneously, these labels can freely overlap on the same text span. For instance, the Spanish imperative verb form *"comprad"* (buy) simultaneously triggers two high-commitment (booster) labels: *Certainty (HC)*, as the imperative mood projects communicative conviction and authority, and *Participation (HC)*, since addressing the audience directly acts as an explicit engagement marker. This multi-label density is essential for capturing the complex pragmatic footprint of natural discourse.

These dimensions are mapped using specific linguistic markers where LLM annotations diverge from human standards (Table 1). Rather than measuring grammatical frequencies, linguistic features, such as verbal moods or pronouns, are treated strictly as vehicles for the underlying pragmatic dimensions of hedging and boosting. Concrete examples of these overlapping markers and their resulting errors are detailed in Section 4 and Appendix A.

Dimension	Linguistic Marker / Trigger
Doubt	Question marks Conditional tense Subjunctive mood Verbs of belief Stressed interrogative pronouns Conditional adverb *"si"* Adverbs indicating possibility
Certainty	Adverb *"sí"* Verbs of knowledge Verb *"ser"* (to be) Verb *"estar"* (to be) Verb *"tener"* (to have) Imperative mood Future tense Periphrases of obligation
Specificity	Numbers and digits Proper names and Named Entities Technical terms
Non-specificity	Generalizing terms Indefinite pronouns Temporal constructions
Participation	First-person verbs Second-person verbs First-person pronouns Second-person pronouns
Distancing	Impersonal reflexive passive Passive voice Third-person possessive pronouns

Table 1: Granular taxonomy of linguistic markers across the six subjectivity dimensions. These structural features serve as the target triggers for identifying and evaluating automated annotation noise.

4 Diagnostic Framework for Label Noise Induced by Generative Models

This section introduces a diagnostic framework to evaluate generative label noise by analyzing discrepancies between automated predictions and the human gold standard (Section 5.2). At its core, this framework operates on a binary codification where each predicted label is classified as either correct or incorrect. The analysis focuses specifically on the incorrect predictions, categorizing them into two primary noise profiles that directly map to standard evaluation metrics:

- **Under-generation (*omissions*):** Occurs when the model fails to detect an existing pragmatic marker (False Negatives), representing the qualitative root of degraded Recall performance.
- **Over-generation (*additions*):** Occurs when the model predicts a marker where none exists in the text (False Positives), representing the qualitative driver behind lower Precision scores.

It is worth clarifying that, although the six subjectivity dimensions are a fixed label set, the words or text spans models must extract for each label are not predefined, so models can generate tokens absent from the source text. When this happens, the predicted span does not exist in the source and is counted as an addition (False Positive), regardless of its assigned label. This may indicate hallucination, but characterizing it falls outside this framework's scope, which focuses only on whether errors manifest as omissions or additions.

Table 2 illustrates an operational example of how these alignments and divergencies are identified.

While standard evaluations rarely classify synthetic data errors using detailed taxonomies (Klie et al., 2023), this methodology goes further by linking each omission and addition to the specific syntactic or pragmatic marker that triggered it, as detailed in Table 1. This word- and span-level mapping enables a highly specialized diagnostic of Spanish subjectivity detection.

This omission/addition framework was not adopted from an external taxonomy, but developed for this study by mapping the observed annotation discrepancies directly onto standard evaluation metrics. This binary error codification follows established practice in annotation error detection (Dickinson and Meurers, 2003; Klie et al., 2023),

Linguistic Marker	Gold Standard	Model Output	Result
Verb of belief	*sospecha* (suspects)	*sospecha* (suspects)	Match
Conditional	*podría* (could)	*podría* (could)	Match
Subjunctive	*hubiera* (had been)	*hubiera* (had been)	Match
Non-marker	**X**	*pudieron* (they could)	Addition
Subjunctive	*hagan* (they do)	**X**	Omission
Conditional	*deberían* (should)	**X**	Omission

Table 2: Examples of error classification for the *Doubt* dimension. The table shows how linguistic triggers are classified as a match, omission, or addition when compared against the gold standard.

which favors empirically grounded error categorization over fixed *a priori* schemes when diagnosing task-specific noise.

Although this marker-level taxonomy is tailored to subjectivity analysis, the underlying diagnostic framework remains highly adaptable: since it fundamentally relies on identifying matches, omissions, and additions at the word or span level, it can be extended to other multi-label tagging tasks by simply mapping the *omission/addition* categories to task-specific features.

5 Experimental Setting

The experimental framework is designed to evaluate LLM-based subjectivity annotation in Spanish, utilizing a traditional machine learning pipeline as a baseline control. This section details the evaluation corpus (Section 5.1), outlines the curation of the human gold standard (Section 5.2), and describes the implementation of both the traditional control (Section 5.3) and the generative LLM configurations (Section 5.4). By relying on Part-of-Speech (PoS) tagging and handcrafted lexicons, this traditional control establishes a stable baseline to benchmark the nature and distribution of generative label noise.

5.1 Dataset

This study uses the *FakeNewsSpanishCorpus* (Posadas-Durán et al., 2019) and *RUN* (Bonet-Jover et al., 2023) datasets, both validated for Spanish disinformation tasks. The selection of these corpora is grounded in the correlation between speaker implication and the spread of unreliable content (Vieira et al., 2020; Jeronimo et al., 2019). These sources cover diverse text formats, including

traditional news, social media, blogs, and independent websites, to capture the varied communicative styles of User-Generated Content (UGC).

For the diagnostic analysis, a high-density evaluation corpus of 52,569 words across 339 sentences is extracted. Although compact, this sample size is a deliberate and statistically justified choice. Based on an estimated amount of 18,000 sentences in the source corpora, this 339-sentence sample yields a 5.3% margin of error at a 95% confidence level ($z = 1.96$, $p = 0.5$). At the token level, this constitutes over 13% of the source text. This scale ensures statistical representativeness while keeping detailed, word-level manual validation feasible. The corpus exhibits a significant annotation density, with nearly one in four words (23.23%) carrying a tagged element related to subjectivity.

5.2 Noise-Free Ground-Truth Standard

To evaluate automated label noise, a gold-standard reference was established through the manual annotation of the complete dataset by an expert linguist. While relying on a single annotator is a structural limitation, this choice is deliberate. Because the objective is to analyze error typologies rather than produce statistical generalizations, removing human inter-annotator variance isolates a highly consistent pragmatic baseline. This noise-free reference is essential to analyze and quantify systemic LLM deviations without introducing confounding variables.

Since this study's objective is to diagnose models' metalinguistic annotation capabilities, not human performance, the gold standard's role is methodological: ensuring that discrepancies in model outputs can be attributed to model behavior rather than reference inconsistency.

5.3 Traditional Lexicon- and Rule-Based Baseline

To evaluate generative models against a traditional control, a rule- and lexicon-based pipeline was implemented using the SpaCy library (Honnibal et al., 2020). This system serves as an interpretable baseline to determine whether label noise stems from the pragmatic complexity of the text or from generative variability.

The pipeline processes Spanish text at the token level through three sequential phases: First, morphological pre-processing applies Part-of-Speech (PoS) tagging to map the grammatical structure. Second, lexicon-based matching targets semantically rich markers; this phase integrates verbs of knowledge and belief extracted from the ADESSE database (García-Miguel et al., 2010)[1], alongside generalization terms and temporal markers. Third, rule-based extraction implements custom heuristics to isolate complex structural patterns, such as standard passive constructions and the Spanish impersonal reflexive passive (utilizing the clitic *'se'*). Finally, overlapping outputs are deduplicated to establish a stable reference for error profiling.

Reporting these three components (PoS tagging, lexicon matching, rule-based extraction) as a unified traditional ML control, rather than in isolation, was a deliberate choice. A prior exhaustive study (Pérez-Montero et al., 2026) determined the optimal ML configuration for this annotation task, showing that spaCy's PoS tagging alone was limited on complex, Spanish-specific phenomena, and that combining it with handcrafted lexicons and rules yielded the strongest performance. This combined pipeline was therefore adopted as the single traditional control, rather than reporting each component separately.

5.4 Evaluated Generative Models

To profile generative label noise across different architectural families, three state-of-the-art models were selected following the criteria of Beguš et al. (2025). These include two proprietary architectures accessed via paid API's, **GPT-5** (OpenAI, 2025) and **Gemini 2.5 Pro** (Gemini Team, 2025), and one open-source model accessed locally via Ollama, **Llama 4 Maverick Instruct** (Team, 2025). For this diagnostic evaluation, all models were queried using default decoding parameters to capture their baseline noise distribution profiles.

5.5 Prompting Strategies and Annotation Task

The automated annotation is framed as a word- and span-level tagging task. Models are required to identify and extract the exact tokens or multi-word expressions that function as the active linguistic markers for each subjectivity dimension (as detailed in Table 1). Because the target corpus is in Spanish, both the system prompts and the model outputs are processed entirely in Spanish to pre-

[1]The University of Vigo's Database of Verbs, Diathesis Alternations, and Syntactic-Semantic Schemas of Spanish (*Base de datos de Verbos, Alternancias de Diátesis y Esquemas Sintáctico-Semánticos del Español*), available at: https://adesse.webs.uvigo.es/.

serve the pragmatic and syntactic nuances of the language.

Following the taxonomy of Liu et al. (2023) and the TELeR hierarchy (Karmaker Santu and Feng, 2023), three prompting configurations were engineered to examine how context layout triggers or mitigates omission and addition errors. All strategies share an identical system instruction that primes the models as expert linguists, isolating pragmatic reasoning from general task compliance. The configurations are structured as follows:

- **Decomposition (TELeR Level 2-3):** A direct, zero-shot directive targeting structural grammatical extractions.
- **Composition (TELeR Level 4):** A step-by-step Chain-of-Thought (CoT) structure enforcing explicit logical checkpoints before label decision.
- **Augmentation (TELeR Level 4-5):** A few-shot learning framework fueled with exemplary context-label sentences.

The full Spanish prompt templates and annotation examples are provided in Appendix B. Since this work is an exploratory analysis, one of whose objectives is to evaluate models' metalinguistic capacity, the complete annotation guidelines were not provided in the prompts. Instead, different prompts were tailored to analyze how models annotate subjectivity under different prompting approaches with varying levels of information, following the Liu et al. (2023) taxonomy. The human annotator followed the full annotation guidelines, as they were also its author; these guidelines are included in Appendix A only to aid readability and are not directly comparable to the prompts used for the models.

6 Diagnostic Results and Noise Analysis

To characterize generative label noise, all automated annotations are evaluated against the human gold standard (Section 5.2). This analysis is structured across three diagnostic layers. First, a macro-level evaluation uses Precision, Recall, and F1-score to establish an overall performance baseline (Subsection 6.1). Second, a granular analysis maps the density of omission and addition errors across each subjectivity dimension (Figure 1). Finally, a qualitative characterization examines how specific linguistic features trigger systematic label noise in both the traditional control and the generative configurations (Subsection 6.2).

6.1 Macro-Level Quantitative Analysis

Within this initial evaluation, **Precision** measures the mitigation of addition errors (false positives), **Recall** reflects the avoidance of omission errors (false negatives), and the **F1-score** measures the overall alignment. These baseline results across all configurations are detailed in Table 3.

Automatic Annotation Noise Profiles. The results reveal distinct label noise patterns across the evaluated systems. The traditional ML control minimizes addition errors (achieving high precision) in low-commitment (LC) categories, such as *Doubt* ($F_1 = 0.78$), *Non-specificity* ($F_1 = 0.80$), and *Distancing* ($F_1 = 0.88$). Conversely, generative LLMs reduce omission errors (yielding higher recall) in high-commitment (HC) categories, such as *Certainty*, *Specificity*, and *Participation*. Among these, Gemini minimizes overall noise, exceeding an F_1 of 0.89 across all HC dimensions while balancing additions and omissions. GPT configurations yield higher systematic noise, peaking at a lower F1-score of 0.63 in *Specificity*. Llama effectively controls omissions in *Specificity* ($F_1 = 0.94$ under the Composition prompt), though it generally produces more addition noise than Gemini.

Impact of Prompt Configurations on Noise Stability. Prompt engineering affects noise generation unevenly across architectures. While GPT and Gemini exhibit stable alignment profiles (with F1 variance of $\Delta F_1 < 0.10$ and < 0.20, respectively), Llama shows high susceptibility to prompt variations. In *Participation* and *Distancing*, Llama's F1-scores fluctuate by up to 0.46, demonstrating that prompt formulation heavily dictates performance stability in open-source models.

Statistical Validation of Noise Distribution. Dimensions with concrete structural markers, such as *Specificity*, consistently generate lower label noise. This suggests that explicit linguistic elements (e.g., named entities) are more resilient to automated error. To validate these patterns, a Friedman test (Friedman, 1937) was applied to the F_1 results. The test confirms that performance variations across models, prompts, and categories are statistically significant ($Q = 31.85, p < 0.001$), supporting the need for a granular qualitative diagnostic of these error patterns.

	Doubt (LC)			Non-specificity (LC)			Distancing (LC)		
Model_Configuration	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Traditional ML Control	**0.72**	**0.84**	**0.78**	**0.77**	**0.84**	**0.80**	**0.88**	0.88	**0.88**
GPT_Decomposition	0.31	0.40	0.35	0.30	0.41	0.35	0.37	0.49	0.43
GPT_Composition	0.28	0.32	0.30	0.28	0.35	0.31	0.36	0.38	0.37
GPT_Augmentation	0.32	0.37	0.34	0.14	0.25	0.18	0.32	0.37	0.34
Gemini_Decomposition	0.40	0.53	0.46	0.47	0.80	0.59	0.39	**0.99**	0.56
Gemini_Composition	0.43	0.56	0.49	0.47	0.81	0.59	0.51	0.97	0.67
Gemini_Augmentation	0.43	0.55	0.48	0.49	0.79	0.61	0.65	0.90	0.76
Llama_Decomposition	0.31	0.40	0.35	0.26	0.61	0.37	0.25	0.32	0.28
Llama_Composition	0.20	0.25	0.22	0.32	0.82	0.46	0.33	0.72	0.46
Llama_Augmentation	0.23	0.30	0.26	0.18	0.77	0.30	0.69	0.77	0.73

	Certainty (HC)			Specificity (HC)			Participation (HC)		
Model_Configuration	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
Traditional ML Control	0.46	0.48	0.47	0.69	0.84	0.76	0.92	0.93	0.93
GPT_Decomposition	0.31	0.40	0.35	0.57	0.58	0.58	0.47	0.51	0.49
GPT_Composition	0.28	0.39	0.33	0.62	0.63	0.63	0.49	0.52	0.50
GPT_Augmentation	0.33	0.46	0.39	0.59	0.61	0.60	0.41	0.44	0.42
Gemini_Decomposition	0.78	0.85	0.81	0.94	**0.99**	**0.97**	0.95	0.95	0.95
Gemini_Composition	0.84	0.88	0.86	**0.94**	0.97	0.96	**0.96**	**0.96**	**0.96**
Gemini_Augmentation	**0.87**	**0.90**	**0.89**	0.80	0.99	0.88	0.93	0.94	0.94
Llama_Decomposition	0.32	0.45	0.37	0.81	0.91	0.86	0.70	0.72	0.71
Llama_Composition	0.22	0.28	0.25	0.92	0.95	0.94	0.59	0.62	0.61
Llama_Augmentation	0.26	0.43	0.32	0.75	0.93	0.83	0.20	0.36	0.25

Table 3: Noise susceptibility benchmarks (Precision, Recall, and F1-score) across the six multi-dimensional scales. Bold values indicate optimal configuration boundaries against synthetic label noise injection.

6.2 Qualitative Analysis of Error Profiles

To analyze the linguistic roots of automated label noise, individual model errors are mapped directly to the proposed diagnostic taxonomy (Section 4). This qualitative profiling isolates specific instances of omission and addition errors.

Table 4 shows divergent error profiles across the evaluated systems. The traditional ML control generates errors predominantly through omissions (63.99%). In contrast, generative models exhibit a pronounced tendency toward addition errors. Within the LLM configurations, Gemini and Llama are heavily biased toward additions (exceeding 51%), whereas GPT produces a high rate of omissions (exceeding 70%) alongside lower overall match rates.

To examine these error patterns across specific linguistic contexts, Table 5 details the distribution at the dimension level, using the optimal prompt configuration for each model family (Composition for GPT and Llama; Augmentation for Gemini).

Low-Commitment Dimensions (LC). This category groups the dimensions where speaker stance is attenuated or references are detargeted: *Doubt*, *Non-specificity*, and *Distancing*.

- **Doubt:** This dimension generates high omission noise in generative models. GPT and Llama produce substantial overall error (72.16% and 79.59%, respectively), primarily driven by omissions (81.43% in GPT). While the traditional control yields a balanced 50/50 distribution, LLMs frequently overlook modal markers (e.g., the Spanish subjunctive) unless explicit lexical indicators are present.
- **Non-specificity:** Polysemous indefinite terms often trigger addition errors in LLMs. While the traditional control achieves a 76.53%

			Noise Type (Relative %)	
Experimental Setup	**% Match**	**% Error**	**% Omit**	**% Add**
Traditional ML Control	72.26	27.74	63.99	36.01
GPT_Decomposition	41.50	58.50	72.81	27.19
GPT_Composition	42.36	57.64	79.20	20.80
GPT_Augmentation	38.78	61.22	70.97	29.03
Gemini_Decomposition	63.96	36.04	18.00	82.00
Gemini_Composition	69.84	30.16	23.86	76.14
Gemini_Augmentation	71.81	28.19	28.69	71.31
Llama_Decomposition	48.60	51.40	48.35	51.65
Llama_Composition	48.84	51.16	35.09	64.91
Llama_Augmentation	42.31	57.69	28.12	71.88

Table 4: Global distribution of label noise across experimental configurations. Noise types describe the relative ratio of omission (% Omit) versus addition (% Add) errors calculated over the total noise (Error) (100%).

Optimized Setup	% Match	% Error	Noise Type (Relative %)	
			% Omit	% Add
Doubt (LC)				
Traditional ML Control	72.48	27.52	50.00	50.00
GPT_Composition	27.84	72.16	81.43	18.57
Gemini_Augmentation	42.96	57.04	60.49	39.51
Llama_Composition	20.41	79.59	76.92	23.08
Certainty (HC)				
Traditional ML Control	46.04	53.96	94.04	5.96
GPT_Composition	28.04	71.96	60.00	40.00
Gemini_Augmentation	**87.11**	12.89	74.68	25.32
Llama_Composition	22.27	77.73	73.28	26.72
Non-specificity (LC)				
Traditional ML Control	76.53	23.47	61.96	38.04
GPT_Composition	28.39	71.61	72.52	27.48
Gemini_Augmentation	49.20	50.80	25.09	74.91
Llama_Composition	31.68	68.32	10.09	89.91
Specificity (HC)				
Traditional ML Control	68.99	31.01	42.47	57.53
GPT_Composition	62.47	37.53	96.05	3.95
Gemini_Augmentation	80.33	19.67	3.76	96.24
Llama_Composition	**91.76**	8.24	54.44	45.56
Distancing (LC)				
Traditional ML Control	88.36	11.64	100.00	0.00
GPT_Composition	35.52	64.48	91.10	8.90
Gemini_Augmentation	65.06	34.94	20.65	79.35
Llama_Composition	33.26	66.74	19.12	80.88
Participation (HC)				
Traditional ML Control	92.44	7.56	94.12	5.88
GPT_Composition	48.84	51.16	87.50	12.50
Gemini_Augmentation	**93.33**	6.67	86.67	13.33
Llama_Composition	59.34	40.66	87.84	12.16

Table 5: Granular noise diagnostics partitioned by dimension under optimal prompting configurations. Noise types describe the relative ratio of omission (% Omit) versus addition (% Add) errors.

match rate, Llama underperforms, generating 89.91% of its errors as additions. High-frequency indefinites (e.g., *todo*, *alguien*) act as noise triggers, causing models to over-annotate without sufficient context analysis.

- **Distancing:** The Spanish impersonal reflexive (*pasiva refleja* with the clitic *'se'*) reveals contrasting behaviors. The traditional control incurs zero additions, maintaining a low overall error rate (11.64%). Conversely, Gemini Augmentation yields 79.35% addition errors, over-interpreting the grammatical marker *'se'* as a subjective perspective marker even in objective contexts.

High-Commitment Dimensions (HC). This category encompasses dimensions featuring assertive speaker commitment and explicit anchoring: *Certainty*, *Specificity*, and *Participation*.

- **Certainty:** Distinct error dynamics emerge here. The traditional control is highly conservative, showing almost exclusive omission noise (94.04%) and nearly zero additions (5.96%). Gemini Augmentation minimizes overall noise to 12.89%, but shifts the bias toward additions (25.32%). This indicates that mitigating omissions in LLMs often introduces synthetic addition noise by over-annotating factual sentences as certain.
- **Specificity:** Explicit syntactic markers (e.g., named entities, numbers) suppress omission noise. Llama Composition achieves the lowest overall noise density (8.24%). However, the residual noise in Gemini Augmentation is almost entirely composed of additions (96.24%), reflecting a systematic over-annotation of objective referential tokens.
- **Participation:** Gemini Augmentation achieves the highest overall alignment, limiting total noise to 6.67%. The error profiles are highly polarized: the traditional control and GPT generate noise almost exclusively via omissions (94.12% and 87.50%, respectively), whereas Gemini balances additions and omissions, demonstrating a robust capability to process explicit personal pronouns and verbal inflections.

Overall Noise Bias Analysis. Figure 1 reports the error bias distribution for each model configuration and dimension, computed as the normalized difference between omission and addition error proportions, ranging from $+1$ (all errors are omissions) to -1 (all errors are additions), with values near 0 indicating a balanced mix. The traditional control and GPT cluster toward positive values across most dimensions (e.g., traditional control: $+1.00$ in *Distancing* and *Participation*; GPT_Augmentation: $+1.00$ in *Doubt*), confirming a systematic failure to detect subjective indicators. Conversely, Gemini shifts toward negative values in *Specificity* (Gemini_Augmentation: -0.56) and *Distancing* (Gemini_Decomposition: -1.00), reflecting a tendency to over-generate fake markers, while Llama shows a more mixed profile in *Specificity*, ranging from -0.20 (Decomposition) to $+0.71$ (Composition). *Participation* shows the strongest and most consistent omission bias across generative models ($+1.00$ for Gemini across all three prompting strategies), while *Doubt* remains predominantly omission-biased across most setups, with Gemini as the exception, showing a mild ad-

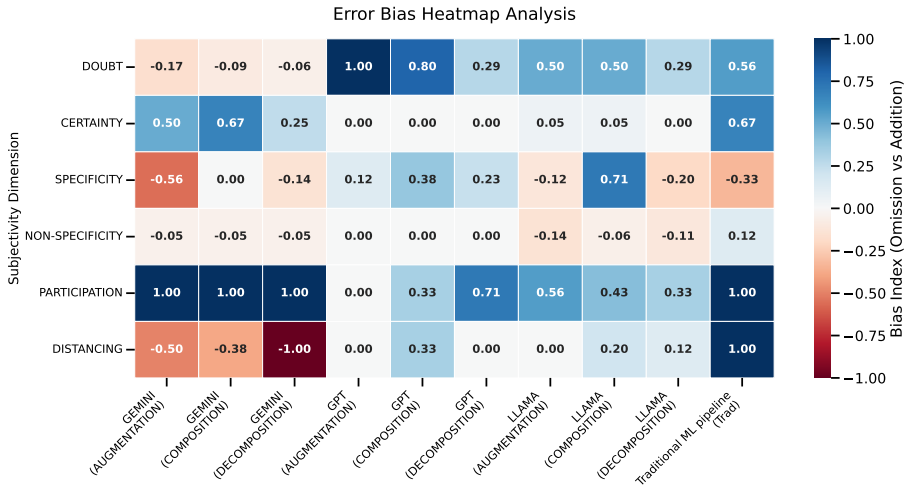


Figure 1: Error bias distribution across categories and model configurations. Positive values (blue) indicate a tendency toward omissions, while negative values (red/orange) indicate a tendency toward additions.

dition bias across all its configurations (-0.17 to -0.06).

7 Conclusion and Future Work

This study presents a diagnostic framework to analyze, categorize, and benchmark label noise in automated text annotation. Beyond standard macro-performance metrics, this approach demonstrates how contextual and grammatical ambiguity translates into systematic error profiles when employing LLMs as automated annotators.

The findings indicate distinct error patterns dictated by system architecture. While the traditional ML control predominantly generates omission errors, generative language models exhibit a pronounced tendency to introduce addition errors when annotating complex subjectivity dimensions. These discrepancies are particularly severe in Spanish structures, such as subjunctive alternations and the impersonal reflexive passive, where automated systems show decreased stability in label assignment. This instability introduces significant label noise, underscoring the necessity of human-in-the-loop validation before LLMs can be reliably adopted as automated annotators.

Taken together, the dimension-level analysis (Section 6) shows that models' metalinguistic capacity is variable, not uniformly strong or weak. Performance improves on dimensions with explicit referential markers, such as numbers or named entities in *Specificity*, which are easy to identify regardless of context. In contrast, performance drops on dimensions that depend on more complex grammatical resolution, most notably *Doubt*, where cues like the subjunctive mood are harder to detect reliably. Crucially, this same pattern appears in the traditional ML control, not only in LLMs, suggesting that the difficulty lies in the subjective nature of the annotation task itself rather than in the metalinguistic capacity of the models alone, given that annotation disagreement is not always an error but may reflect legitimate diversity in interpretation (Piot et al., 2026).

Consequently, the proposed taxonomy provides a structured methodology to audit automated dataset reliability. Future work will mainly focus on expanding this diagnostic approach to multi-annotator environments, as well as performing further analysis on the applicability of subjectivity-aware evaluation metrics that can handle the variability reported in previous work (Piot et al., 2026; Sharma et al., 2026). Additional research lines could include engineering prompt strategies specifically designed to mitigate addition errors, exploring repeated-querying annotation protocols instead of single-shot predictions to improve consistency on subjective tasks to generate coherent annotations that reduce noise (Sharma et al., 2026), and validating the stability of these synthetic noise profiles across other typologically diverse languages.

Limitations

This study presents three primary limitations. First, although the dataset size is constrained, this scale is a deliberate trade-off enabling the exhaustive, word-level manual analysis required to profile label noise. Conducting a manual validation of this granularity on substantially larger corpora would

be logistically unfeasible.

Second, the observed error distributions are closely tied to Spanish subjectivity markers (Klie et al., 2023), such as subjunctive alternations and the clitic *"se"*. While these specific patterns are language-dependent, the underlying diagnostic taxonomy remains highly generalizable and can be adapted to analyze label noise in other linguistic contexts.

Finally, this evaluation is limited to a specific snapshot of large language models and prompting strategies. While model updates will inherently alter absolute noise baselines over time, the proposed methodology establishes a reproducible framework to benchmark, track, and audit automated annotation workflows.

Ethical Considerations

This study utilizes exclusively public, anonymized, and benchmarked datasets, avoiding direct data collection. Additionally, this research focuses solely on characterizing automated label noise to improve dataset quality, without generating synthetic content or promoting harmful applications.

Acknowledgments

This research is funded by a grant for the recruitment of predoctoral research staff (CIACIF/2023/106) from the Fondo Social Europeo Plus of Generalitat Valenciana - European Social Fund Plus of the Generalitat Valenciana. The research work is part of the R&D projects; "SAFEWORDS": Language Anonymization with Ethical and Legal Safeguards through NLP (AIA2025-163322-C63); "Mecánica cuántica para la comprensión y generación del lenguaje" (PID2024-160791OB-I00) funded by MICIU/AEI/10.13039/501100011033 and by FEDER, UE; Proyecto Desarrollo de Modelos ALIA within the framework of the Plan Nacional de Tecnologías de Lenguaje -ENIA 2024 del Ministerio para la Transformación Digital y de la Función Pública y PRTR, NextGeneration EU, Resol. SEDIA 19.08.2024; "Criterios de Evaluación para Corpus de Calidad en Inteligencia Artificial (CRITERIA)", developed in the II Concurso Nacional para la adjudicación de Ayudas a la Investigación en Humanidades 2025, with the topic "Humanidades Digitales" (Referencia: FRAHUMANIDADES25-01), funded by Fundación Ramón Areces.

References

Mostafa M. Amin, Rui Mao, Erik Cambria, and Bjorn W. Schuller. 2024. A Wide Evaluation of ChatGPT on Affective Computing Tasks . *IEEE Transactions on Affective Computing*, 15(04):2204–2212.

Francesco Antici, Luca Bolognini, Matteo Antonio Inajetovic, Bogdan Ivasiuk, Andrea Galassi, and Federico Ruggeri. 2021. Subjectivita: An italian corpus for subjectivity detection in newspapers. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings 12*, pages 40–52. Springer.

Mijail Bajtin. 1982. *Estética de la creación verbal*. Siglo XXI Editores.

Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment analysis is not solved! assessing and probing sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, page 12–23. Association for Computational Linguistics.

Gašper Beguš, Maksymilian Dąbkowski, and Ryan Rhodes. 2025. Large linguistic models: Investigating llms' metalinguistic abilities. *IEEE Transactions on Artificial Intelligence*, 6(12):3453–3467.

Divya Bhadauria, Alejandro Sierra Múnera, and Ralf Krestel. 2024. The effects of data quality on named entity recognition. In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, page 79–88. Association for Computational Linguistics.

Alba Bonet-Jover, Robiert Sepúlveda-Torres, Estela Saquete, Patricio Martínez-Barco, Alejandro Piad-Morffis, and Suilan Estevez-Velarde. 2023. Applying human-in-the-loop to construct a dataset for determining content reliability to combat fake news. *Engineering Applications of Artificial Intelligence*, 126:107152.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Markus Dickinson. 2015. Detection of annotation errors in corpora. *Language and Linguistics Compass*, 9(3):119–138.

Markus Dickinson and W. Detmar Meurers. 2003. Detecting errors in part-of-speech annotation. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics - EACL '03*, volume 1 of *EACL '03*, page 107. Association for Computational Linguistics.

Allyson Ettinger, Jena Hwang, Valentina Pyatkin, Chandra Bhagavatula, and Yejin Choi. 2023. "you are an expert linguistic annotator": Limits of llms as analyzers of abstract meaning representation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, page 8250–8263. Association for Computational Linguistics.

Milton Friedman. 1937. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701.

José M. García-Miguel, Gael Vaamonde, and Fita González Domínguez. 2010. ADESSE, a database with syntactic and semantic annotation of a corpus of Spanish. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Google Gemini Team. 2025. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*. Release date derived from search results: July 2025.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spacy: Industrial-strength natural language processing in python.

Ken Hyland. 2005. Metadiscourse: Exploring interaction in writing. *Continuum*.

Maliha Jahan, Helin Wang, Thomas Thebaud, Yinglun Sun, Giang Ha Le, Zsuzsanna Fagyal, Odette Scharenborg, Mark Hasegawa-Johnson, Laureano Moro Velazquez, and Najim Dehak. 2024. Finding spoken identifications: Using gpt-4 annotation for an efficient and fast dataset creation pipeline. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, LREC-COLING, page 7296–7306. European Language Resources Association (ELRA) and ICCL.

Caio Libanio Melo Jeronimo, Leandro Balby Marinho, Claudio E. C. Campelo, Adriano Veloso, and Allan Sales da Costa Melo. 2019. Fake news classification based on subjective language. In *Proceedings of the 21st International Conference on Information Integration and Web-based Applications amp; Services*, iiWAS2019, page 15–24. ACM.

Shubhra Kanti Karmaker Santu and Dongji Feng. 2023. Teler: A general taxonomy of llm prompts for benchmarking complex tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, page 14197–14203. Association for Computational Linguistics.

Emaliana Kasmuri and Halizah Basiron. 2017. Subjectivity analysis in opinion mining—a systematic literature review. *Int J Adv Soft Comput Appl*, 9(3):132–159.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2023. Annotation error detection: Analyzing the past and present for a more coherent future. *Computational Linguistics*, 49(1):157–198.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

María Antonia Martín Zorraquino. 1999. Aspectos de la gramática y de la pragmática de las partículas de modalidad en español actual. In María Carmen Losada Aldrey, José F. Márquez Caneda, and Tomás Eduardo Jiménez Juliá, editors, *Español como lengua extranjera, enfoque comunicativo y gramática: actas del IX congreso internacional de ASELE. Santiago de Compostela, 23-26 de septiembre de 1998*, pages 25–56. Santiago de Compostela.

Arbi Haza Nasution and Aytuğ Onan. 2024. Chatgpt label: Comparing the quality of human-generated and llm-generated annotations in low-resource language nlp tasks. *IEEE Access*, 12:71876–71900.

OpenAI. 2025. GPT-5 Technical Report. Blog post or Technical Report (OpenAI). Release/announcement date: August 2025.

Alba Pérez-Montero, Elena Lloret Pastor, and Paloma Moreda Pozo. 2026. Hybrid annotation framework for spanish subjectivity detection. *Procesamiento del Lenguaje Natural*, 76:135–148.

Paloma Piot, David Otero, Patricia Martin-Rodilla, and Javier Parapar. 2026. Can llms evaluate what they cannot annotate? revisiting llm reliability in hate speech detection. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, LREC, page 4358–4370. European Language Resources Association (ELRA).

Juan-Pablo Posadas-Durán, Helena Gómez-Adorno, Grigori Sidorov, and Jesús Jaime Moreno Escobar. 2019. Detection of fake news in a new corpus for the spanish language. *Journal of Intelligent & Fuzzy Systems*, 36(5):4869–4876.

Hope Schroeder, Deb Roy, and Jad Kabbara. 2025. Just put a human in the loop? investigating llm-assisted annotation for subjective tasks. In *Findings of the Association for Computational Linguistics: ACL 2025*, page 25771–25795. Association for Computational Linguistics.

Neha Sharma, Navneet Agarwal, and Kairit Sirts. 2026. Towards consistent detection of cognitive distortions: Llm-based annotation and dataset-agnostic evaluation. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, LREC, page 10866–10882. European Language Resources Association (ELRA).

Mohammad Shokri, Vivek Sharma, Elena Filatova, Shweta Jain, and Sarah Levitan. 2024. Subjectivity detection in English news using large language models. In *Proceedings of the 14th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 215–226, Bangkok, Thailand. Association for Computational Linguistics.

Siyuan Song, Jennifer Hu, and Kyle Mahowald. 2025. Language models fail to introspect about their knowledge of language. *Preprint*, arXiv:2503.07513.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, and 431 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. Featured Certification.

Meta Llama Team. 2025. Llama 4 Models: Multimodal, Multilingual, and Mixture-of-Experts Architecture. Technical Report/Official Documentation (Hugging Face / Llama Website). Llama 4 Maverick 17B-128E Instruct. Release date: April 2025.

Lucas Lima Vieira, Caio Libanio Melo Jeronimo, Claudio E. C. Campelo, and Leandro Balby Marinho. 2020. Analysis of the subjectivity level in fake news fragments. In *Proceedings of the Brazilian Symposium on Multimedia and the Web*, WebMedia '20, page 233–240, New York, NY, USA. Association for Computing Machinery.

Maria C. Voutsa, Nicolas Tsapatsoulis, and Constantinos Djouvas. 2025. Biased by design? evaluating bias and behavioral diversity in llm annotation of real-world and synthetic hotel reviews. *AI*, 6(8):178.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*, HLT '05, page 347–354. Association for Computational Linguistics.

Shanshan Xu, Santosh T.y.s.s, and Barbara Plank. 2026. Position: From noise to signal to selbstzweck - reframing human label variation in the era of post-training in nlp. In *Findings of the Association for Computational Linguistics: ACL 2026*, page 23762–23772. Association for Computational Linguistics.

A Annotation Scheme: Dimensions and Linguistic Markers

Table 6 presents the automatic annotation framework, outlining the linguistic indicators and examples evaluated for each dimension.

B Prompting Configuration Examples

This appendix presents the templates for the prompting configurations discussed in Section 5.5. To preserve the original contextual features, all instructions were executed in Spanish. The following examples use the *Doubt* dimension as a baseline reference to illustrate the structural variations across the evaluation settings.

- **Decomposition**. The instruction uses a direct, zero-shot directive for marker extraction.

  <*You are an expert linguist. Your goal is to extract words, sets of words, or punctuation marks that indicate DOUBT, as expressed verbally through Rhetorical Questions (question marks), Conditional Verbs, Subjunctive Mood Verbs (..). Analyze the following text and return the annotated text, marking the extracted text fragment using angle brackets '<>' and the dimension DOUBT using square brackets '[]'. The text to analyze is as follows: [Insert text here].*>

- **Composition**. The instruction uses a step-by-step chain-of-thought structure to guide the annotation process.

  <*You are an expert linguist. Your objective is to extract words, sets of words, or punctuation marks that indicate DOUBT; to achieve this, you must follow six steps: Step 1: Search for question marks (¿?); Step 2: Locate conditional verbs (e.g., "sería" [would be], "podría" [could]); Step 3: Detect subjunctive mood verbs (e.g., "dudara" [I/he/she doubted], "sospechara" [I/he/she suspected]); Step 4: Mark verbs of belief based on their meaning (e.g., "cuestionar" [to question], "desconfiar" [to mistrust]); Step 5: Extract interrogative pronouns (e.g., "qué" [what], "cómo" [how]); and finally, Step 6: Analyze the text and return the annotated text as output, marking the extracted text fragment using angle brackets '<>' and the dimension DOUBT using square brackets '[]', utilizing the following text for analysis: [Insert text here].*>

- **Augmentation**. The instruction includes in-context examples from the corpus to guide the automatic labeling process (few-shot learning).

  <*You are an expert linguist. Your objective is to annotate words, word groups, or punctuation marks that*

Dimension	Linguistic Markers	Examples (Spanish / English)
Degree of Certainty		
Doubt (LC)	- Conditional and subjunctive - Modal adverbs (*quizá*) - Verbs of belief (*creer*)	- "*Quizá* sea tarde" (*Maybe it's late*) - "*Creo* que *podría* funcionar" (*I think it could work*) - "*Parece* que lloverá" (*It seems it will rain*)
Certainty (HC)	- Affirmative adverbs (*sí*) - Future tense (conviction) - Verbs of knowledge	- "*Sé* que vendrá" (*I know he/she will come*) - "Es *evidente* el cambio" (*The change is evident*) - "Esto *garantizará* el éxito" (*This will guarantee success*)
Level of Concreteness		
Non-specificity (LC)	- Generalizing terms - Indefinite pronouns - Vague quantifiers	- "*Siempre* dicen lo mismo" (*They always say the same*) - "*Alguien* debería actuar" (*Someone should act*) - "*Todo* el mundo lo sabe" (*Everyone knows it*)
Specificity (HC)	- Proper names and acronyms - Specific dates - Technical entities	- "Hablamos con *García*" (*We spoke with García*) - "Ocurrió en *Madrid*" (*It happened in Madrid*) - "El informe de la *ONU*" (*The UN report*)
Scale of Involvement		
Distancing (LC)	- Impersonal reflexive (*se*) - Passive voice - 3rd person plural agency	- "*Se dice* que..." (*It is said that...*) - "*Fue decidido* ayer" (*It was decided yesterday*) - "*Comentan* que hay problemas" (*They say there are problems*)
Participation (HC)	- 1st/2nd person pronouns - Possessives - Direct address	- "*Mi* opinión es..." (*My opinion is...*) - "*Queremos* informarte" (*We want to inform you*) - "¿Qué opinas *tú*?" (*What do you think?*)

Table 6: Annotation framework categorized by dimension, linguistic markers, and translated examples.

indicate DOUBT, as expressed verbally through question marks, Conditional Verbs, Subjunctive Mood Verbs, (...). The reference examples are: 1. "Si fuera natural, solo habría afectado a países con la misma temperatura que China" (where 'Si' [If] indicates conditional subordination, 'fuera' [were] is subjunctive, and 'habría' [would have] is conditional) (...). Perform the text analysis and return the annotated text as output, marking the extracted text fragment using angle brackets '<>' and the dimension DOUBT using square brackets '[]', utilizing the following text for analysis: [Insert text here].>

Increasing the Robustness of the Fine-tuned Multilingual Machine-Generated Text Detectors

Dominik Macko, Robert Moro, Ivan Srba
Kempelen Institute of Intelligent Technologies, Slovakia
{dominik.macko, robert.moro, ivan.srba}@kinit.sk

Abstract

Since the proliferation of LLMs, there have been concerns about their misuse for harmful content creation and dissemination. Recent studies justify such fears, providing evidence of LLM vulnerabilities and high misuse potential. It has been shown that humans struggle to reliably distinguish high-quality machine-generated text from authentic human-written text. Therefore, it is crucial to develop automated means to accurately detect machine-generated content. While multiple detectors of machine-generated texts have been researched so far, a known issue is their low generalizability to unseen generators, domains or languages. This work[1] addresses the problem by proposing a robustness-oriented LLM training methodology for the detection task, making the detectors more robust against obfuscation and more generalizable to out-of-distribution data. The fixed-size ablation study confirms that the gains are not simply a function of more data.

1 Introduction

The number, variety, enhanced capabilities, and simple usage of large language models (LLMs) cause growing concerns about a massive generation of harmful content with LLMs being potentially maliciously misused (Borji, 2023; Zhuo et al., 2023; Crothers et al., 2023). Previous research has demonstrated the ability of LLMs to produce disinformation texts (Vykopal et al., 2024; Lucas et al., 2023), even personalized ones (Zugecova et al., 2025; Gabriel et al., 2024), frauds for academic exams (Achiam et al., 2023), or plagiarism (Wahle et al., 2022). The fears of misuse are more severe due to indistinguishability between human-written texts and high-quality machine-generated texts (MGTs) (Sadasivan et al., 2025). Thus, the MGT detection (MGTD) belongs to the key challenges associated with state-of-the-art LLMs as identified by Kaddour et al., 2023.

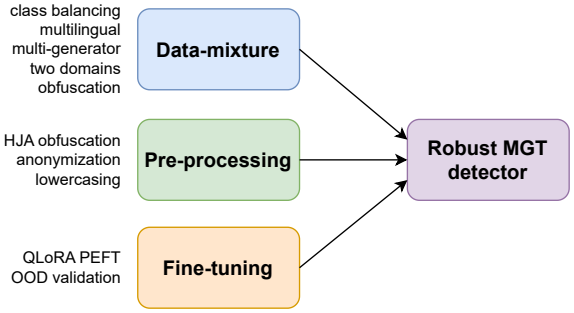


Figure 1: Overview of the proposed robustness-oriented MGT detector training methodology.

Since the harmful online content, such as misinformation, disinformation, propaganda, or impersonation, can be also generated or adjusted by LLMs, its automated detection can crucially help to fight such phenomena. Indication that a potentially harmful content is generated by using artificial intelligence (AI) can alleviate its eventual effect on people. By its statistical nature, the AI generated content should be considered less credible (in comparison to authentic human-written texts), mainly due to the simplicity and speed of creating the content as well as a potential presence of hallucinations (unintended misleading information).

However, the state-of-the-art MGTD methods suffer from performance degradation in out-of-distribution (OOD) settings, such as new generators, domains, languages, and paraphrasing or adversarial attacks (Antoun et al., 2023; Li et al., 2024). Therefore, increasing their robustness in this manner is the primary concern. Ensuring that some data are really OOD (in relation to MGTD training data) is quite challenging since even languages are related to each other differently (language families, writing scripts), the same regarding generators (architectural similarities, training data), and other aspects as well. However, there exist

[1]The source code available at: https://github.com/kinit-sk/robust_mgt_detector

Proceedings of the 11th Workshop on Natural User-generated Text, pages 154–166
October 28, 2026 ©2026 Association for Computational Linguistics

measures of similarities between data distributions, such as MAUVE score (Pillutla et al., 2021).

This work addresses this problem by mixing the training data in a way to include multiple languages, domains, generators and even obfuscated (attack) data, along with multiple unique pre-processing steps (see Figure 1). We specifically target these research questions:

RQ1 *Can multilingual MGT detectors be made more robust to OOD data through diverse fine-tuning data mixtures?*

RQ2 *How do different foundation model architectures and scales behave under robust MGTD fine-tuning methodology?*

To the best of our knowledge, this is the first attempt to systematically fine-tune robust multilingual MGT detectors. Although such detectors manifest a small performance degradation on in-distribution data, the OOD performance is significantly increased, making them more suitable to be deployed in the real-world settings.

The key contributions of this work include:

- **A systematic methodology for robust fine-tuning** for MGT detection task, using diverse data for training and OOD data for validation, including unique pre-processing steps.
- **Comparison of variety of foundational models** robustly fine-tuned as MGT detectors by evaluation on in-distribution as well as OOD datasets.
- **Controlled evidence that gains are not a pure data-volume artifact**, where fixed-size experiments (32k/100k/1M) and a component-wise ablation (v0–v6) isolate which elements of the methodology drive OOD robustness gains.
- **Introduction of highly diverse mixture dataset** (MIX), along with its 2k-samples subset for validation (MIX2k), covering 7 languages, over 10 domains, and over 50 generators with variable generation settings. We provide the data construction source code for replication purpose (see Appendix B).

2 Related Work

The evaluation of detection robustness is unfortunately focused mostly on a single aspect in existing works. For example, MULTITuDE (Macko et al., 2023) was focused on cross-lingual generalization, MAGE (Li et al., 2024) on cross-domain generalization, and RADAR (Hu et al., 2023) or OUTFOX (Koike et al., 2024) on adversarial robustness.

There are attempts to combine multiple aspects in the evaluation, such as (Dugan et al., 2024; Macko et al., 2024). However, the combined evaluation on completely unrelated OOD data (different generators, languages, domains, generation settings) is still lacking (especially in multilingual scenario). SemEval-2024 Task 8 multilingual dataset (Wang et al., 2024) provides a way to test a sort of multi-generator, multi-domain, and multilingual performance. However, due to inconsistent inclusion of various data (e.g., some domains represented only in a single language), the evaluation might be somewhat biased. Therefore, one should train the detectors on another data and use this dataset only for OOD evaluation.

In comparison to existing works, our approach combines their strengths and eliminates their weaknesses. Specifically, our approach integrates multiple evaluation dimensions considered separately in prior work while enabling unified training and evaluation.

3 Robustness-oriented Detector Training

A natural option for increasing OOD performance is to include all available data in the training. However, it would result in a problem of leaving no data for OOD evaluation. We cope with the problem by combining a wide set of diverse datasets into the MIX evaluation dataset that is used for OOD evaluation. We leave only carefully selected data (unrelated to evaluation data) for training and show how our unique strategy (overviewed in Figure 1 and below) empirically improves the detection robustness.

Data-mixture. Previous works have shown good cross-lingual capabilities (even to unseen languages) of MGT detectors when trained on high amount of languages. Therefore, we cover **the widest set of languages** for the training (over 44). For better generalization of detectors to unseen domains, we must include at least **two domains** in the training for the same generators and languages, to prevent biases to specific pairs. Therefore, we combine social-media texts and news articles in the training (as having distinct writing styles), leaving other domains for OOD evaluation. And finally,

to increase adversarial robustness of detectors, we include **obfuscated texts** (i.e., modified by adversarial attack, paraphrasing, and backtranslation to evade the detection) in the training. More details about the datasets can be found in Section 3.1.

Pre-processing. To further increase the robustness of the detection towards obfuscation, we introduce a combined **HomoglyphJoinerAttack** (see Section 3.2) that is applied on-the-fly to modify a portion of the training data and thus make the detector resistant to homoglyphs (visually similar characters that might confuse humans) and invisible spaces. We further apply **anonymization** and **lower-casing** to the texts, in order the detector to focus on the meaning rather than on visual appearance (and minimize biases towards personal information). More details regarding pre-processing can be found in Section 3.2.

Fine-tuning. For training, we use the LLM fine-tuning for the sequence classification task. Based on the model size, we either use half-precision full fine-tuning, or we use **QLoRA** parameter-efficient fine-tuning approach. We manually tune the hyper-parameters for all tested model-variant to be able to successfully learn the task. We tend to use the same hyper-parameters for all the models for comparison purposes. More details about the process and the used hyper-parameters can be found in Section 3.3. During the fine-tuning process, a small portion of **OOD data is used for validation** – i.e., the selection of model checkpoint that is the most generalizable to OOD.

3.1 Datasets

For training and evaluation purposes, we used a combination of multiple datasets described below.

MultiSocial (Macko et al., 2025) A dataset of real social-media texts in 22 languages from 5 platforms, containing also the generated counterparts (by using 3 iterations of paraphrasing) by 7 generators. We have balanced the train split of this dataset by additional human samples (using the published procedure[2]), covering also languages out of the intended 22 (specifically, 40 languages with at least 1,000 samples).

MULTITuDE_v2 (Macko et al., 2024) A subset of MULTITuDE_v2 dataset of obfuscated MULTITuDE (Macko et al., 2023) texts, covering the three selected authorship obfuscation methods (DFTFooler adversarial attack, ChatGPT paraphrasing, and m2m100-1.2B backtranslation). Due to modification (obfuscation) of the texts by machines, we include only machine-labeled samples (uncertain labeling of machine-obfuscated human texts).

MULTITuDE_v3 (Macko et al., 2025) An extended version of the MULTITuDE (Macko et al., 2023) dataset of real human-written news articles in 22 languages accompanied by counterparts generated by 7 generators, created for OOD evaluation by (Macko et al., 2025). We have balanced their train split of the dataset by additional human samples (using the same published procedure[3]).

SemEval (Wang et al., 2024) Multilingual dataset used during SemEval-2024 Task 8.

MIX Our dataset created by merging, de-duplicating and sampling 100k human and 100k machine samples from 18 existing datasets for binary MGTD task (for more details about the dataset creation process, see Appendix B). It represents the most diverse dataset of this kind created so far.

MIX2k A subset of a MIX dataset, covering various domains, 7 languages and over 50 generators. It contains 1k human and 1k machine samples.

For *training (fine-tuning)* of detectors, we combine the train splits of the first three datasets (Table 1). After de-duplication, the obtained set includes 427,195 human samples and 554,216 machine samples out of 2 domains (news and social media), 44 languages (of more than 1,000 samples), and 16 generators. A small imbalance in human vs machine classes was addressed by up-sampling of the minority class (i.e., human) during the training. The MIX2k dataset (containing mostly OOD data) was used for validation during the fine-tuning, i.e., for checkpoint selection.

For the *in-distribution evaluation*, we use the test splits of the MULTITuDE_v3 and MultiSocial datasets. For the *OOD evaluation*, we use the test

[2] https://github.com/kinit-sk/multisocial

[3] https://github.com/kinit-sk/mgt-detection-benchmark

Dataset	Human	Machine	Generator	Language
MultiSocial	290,618	290,618	7	40
MULTITuDE_v2	0	120,090	7	3
MULTITuDE_v3	136,577	143,527	7	22
MIX2k	1,000	1,000	>50	7

Table 1: Text samples counts for the train splits of the included datasets (MIX2k used for validation only).

Dataset	Human	Machine	Generator	Language
MultiSocial	17,367	121,460	7	22
MULTITuDE_v3	6,502	44,566	7	22
MIX	99,759	100,000	>50	7
SemEval	20,238	22,140	7	4

Table 2: Text samples counts for the test splits of the evaluation datasets.

split of the SemEval, as well as the full MIX dataset. The statistics regarding sample counts for testing are summarized in Table 2.

3.1.1 Data Distribution Evaluation

To measure data distribution similarities (mostly in regard to OOD evaluation), we are calculating MAUVE (Pillutla et al., 2021) scores and Frontier integrals (Liu et al., 2021) between individual train and test sets. MAUVE uses calculation of Kullback–Leibler (KL) divergences between the two data distributions in a quantized embedding space of a foundation model (we employ BERT-base-multilingual-cased). Frontier integral is a statistical summary of divergence frontiers, measuring the quality-diversity trade-off. Baseline train set represents the training data used for baseline fine-tuned detector. Robust train set is the proposed data mixture used for robust detector fine-tuning. For MAUVE, higher values indicate that the distributions are closer to each other. For Frontier integral, the same is indicated by lower values. The results are provided in Table 3. It can be clearly seen that the MultiSocial and MULTITuDE_v3 test splits are closer to training data distributions than the MIX and SemEval test datasets.

3.2 Texts Pre-processing

To increase the robustness against obfuscation methods (preventing the detection), we modify a portion of machine-generated training samples by our new modified attack, combining a generic HomoglyphAttack (introduced by Macko et al., 2024) with insertion of zero-width-joiner (ZWJ) character

Train	Test	MAUVE	Frontier integral
baseline	MultiSocial	0.6493	0.1449
robust	MultiSocial	0.3985	0.2420
baseline	MULTITuDE_v3	0.2131	0.3536
robust	MULTITuDE_v3	0.1852	0.3780
baseline	MIX	0.0268	0.6971
robust	MIX	0.0367	0.6461
baseline	Semeval	0.0071	0.9099
robust	Semeval	0.0086	0.8792

Table 3: Evaluation of data distribution shift between train and test datasets.

```
def homoglyphJoiner(input_text):
  output = ""
  for char in input_text:
    if (random.random() < H_PROBABILITY):
      hcount = len(confusable_characters(char))
      idx = int(random.random()*hcount)
     output += confusable_characters(char)[idx]
    else:
      output += char
    if (random.random() < ZWJ_PROBABILITY):
      output += "\u200D"
  return output
```

Figure 2: HomoglyphJoinerAttack algorithm.

(inspired from GPTZeroBypasser[4]).

This attack, called HomoglyphJoinerAttack (HJA), uses the whole confusables table[5] to psudo-randomly replace letters for their homoglyphs (visually similar characters). Also, it pseudo-randomly inserts a character that is invisible for humans (non-printable). A detailed Python algorithm is provided in Figure 2. We use both, the H_PROBABILITY (a probability of a character to be replaced by its random homoglyph) and ZWJ_PROBABILITY (a probability of insertion of the ZWJ character after the current character of the string) of 0.05. In this way, we have modified about 10% of MGTs of the training set.

In order for the detector to focus on the meaning of the text rather than on appearance and to avoid biases towards specific personal information, we have algorithmically anonymized the texts by replacing the identified email addresses, user mentions, and phone numbers by the corresponding tags of [EMAIL], [USER], [PHONE]. The specific regular expressions we are using to identify such information are available in the enclosed source code. Afterwards, the texts are converted to lowercase as part of the pre-processing. This kind of anonymization and lowercase conversion is done for each text, not only for training but for testing as well.

3.3 Fine-tuning Procedure

We used two kinds of fine-tuning, based on the model size. For smaller models, we employ the full fine-tuning process utilizing half precision, with the default AdamW optimizer with learning rate of 2e−6, batch size of 32, validation each 2,000 steps, running at max for 10 epochs or 4 days. For

[4] https://github.com/jayyt12161/GPTZero-Bypasser/blob/main/GPTZero-Bypasser.py

[5] https://www.unicode.org/Public/security/8.0.0/confusables.txt

the models having more than 2B parameters, we employ QLoRA (Dettmers et al., 2024) parameter efficient fine-tuning process with the weighted cross entropy with logits for loss calculations and 4-bit quantization using BitsAndBytes[6]. The LoRA configuration[7] used an *alpha* of 16, a *dropout* of 0.1, *r* of 64, and the *task type* of sequence classification targeting all linear modules. In comparison to full fine-tuning, the QLoRA fine-tuning version used paged AdamW optimizer with learning rate of $2e-5$, gradient accumulation of 4 steps, and evaluation each 200 steps.

We used these hyper-parameter values for all models to make them comparable (i.e., robust variants use the exact same optimization hyper-parameters as baseline). However, if a smaller batch size was necessary due to memory constraints in case of bigger models, we tuned the values accordingly.

4 Experimental Settings

As a baseline, we use the **mDeBERTa-v3-base** model (as shown by existing works as the best of smaller multilingually pre-trained models, e.g., by Macko et al., 2023) fine-tuned on a combination of MULTITuDE_v3 and MultiSocial datasets. Specificallly, for baseline training, we use their train splits without our balanced extension by additional human samples, i.e., about 486k samples with about 61k being human written (up-sampling of minority class is used for fine-tuning), using 10% of train data for validation. We selected 5 models with diverse architectures and numbers of parameters for robust fine-tuning: 1) mDeBERTa-v3-base (He et al., 2022) to compare the effect of the robust fine-tuning with the same underlying model used as a baseline; 2) Qwen2 (Yang et al., 2024) models with 0.5B and 1.5B parameters to see the effect of model size within the same family; 3) Yi-1.5-6B (AI et al., 2024) to see the effect of bilingually pre-trained model; and 4) Gemma-2-9b-it (Team, 2024) to see the effect of a bigger, instruction fine-tuned model.

As the evaluation metric, we are primarily using **AUC ROC** (area under curve of the receiver operating characteristic) as a classification-threshold independent metric reflecting generic detection capability. For a fine-grained comparison, we are also using TPR@5%FPR, representing true positive rate (TPR) using the classification threshold tuned to reach 5% of false positive rate (FPR) based on ROC curve, and Macro F1 (macro average of the F1 score) reflecting a balance between precision and recall metrics.

[6] https://pypi.org/project/bitsandbytes

[7] https://pypi.org/project/peft

5 Results

When evaluating on in-distribution data (Table 4), we can notice the highest performance of the baseline detector in both AUC ROC and TPR @ 5% FPR metrics in news as well as social-media texts. This indicates that the baseline is a strong in-distribution model. However, we hypothesize that such a baseline, trained purely on the train splits of the evaluation datasets, suffers from over-fitting to the data distribution and thus generalizes worse to OOD data.

All of the detectors are performing better on the MULTITuDE_v3 news articles than on the Multisocial social-media texts, where the lowest performance is achieved by robustly fine-tuned mDeBERTa-v3-base and Qwen2-1.5B models.

When looking at the per-language AUC ROC performance (Table 6), we can see that the detection works well for all the languages evenly, with a small exception of surprise languages (ga, gd, sl) in MultiSocial (i.e., not enough social-media texts for training in some languages). This further **evokes a lower generalizability to OOD** data.

As the results for OOD evaluation indicate (Table 5), the baseline detector is the worst performing on both OOD datasets, which **confirms our hypothesis about over-fitting**. The AUC ROC

	MULTITuDE_v3		MultiSocial	
Detector	**AUC ROC**	**TPR@5%FPR**	**AUC ROC**	**TPR@5%FPR**
Gemma-2-9b-it	0.9914	0.9798	0.9563	0.8400
Qwen2-0.5B	0.9785	0.9316	0.9582	0.8413
Qwen2-1.5B	0.9883	0.9460	0.9549	0.7781
Yi-1.5-6B	0.9748	0.9115	0.9474	0.8008
mDeBERTa	**0.9959**	0.9797	0.9540	0.7750
baseline	0.9944	**0.9875**	**0.9746**	**0.8862**

Table 4: Evaluation of detection performance on in-distribution data. The highest value in each column is boldfaced.

	MIX		SemEval	
Detector	**AUC ROC**	**TPR@5%FPR**	**AUC ROC**	**TPR@5%FPR**
Gemma-2-9b-it	**0.8901**	**0.5227**	**0.9448**	0.8284
Qwen2-0.5B	0.6499	0.0000	0.8434	0.0000
Qwen2-1.5B	0.7588	0.3064	0.9391	**0.8287**
Yi-1.5-6B	0.8167	0.0000	0.8922	0.0000
mDeBERTa	0.6669	0.0502	0.8666	0.6843
baseline	0.5502	0.0000	0.8305	0.0000

Table 5: Evaluation of detection performance on out-of-distribution data. The highest value in each column is boldfaced.

		Test Language [AUC ROC]																					
	Detector	ar	bg	ca	cs	de	el	en	es	et	ga	gd	hr	hu	nl	pl	pt	ro	ru	sk	sl	uk	zh
MULTITuDE_v3	Gemma-2-9b-it	**0.999**	**0.998**	0.988	**1.000**	0.995	**0.997**	**0.997**	0.990	0.959	0.994	0.921	0.996	0.996	0.991	**1.000**	**0.998**	**0.998**	**0.995**	0.998	0.999	**0.996**	0.996
	Qwen2-0.5B	0.958	0.985	0.992	0.995	0.989	0.936	0.980	0.982	0.953	0.974	0.941	0.983	0.975	0.992	0.989	0.996	0.993	0.966	0.992	0.993	0.975	0.988
	Qwen2-1.5B	0.993	0.988	0.997	0.997	0.987	0.925	0.994	0.994	0.963	0.990	0.966	0.991	0.985	0.997	0.994	0.996	0.994	0.985	0.998	0.993	0.982	**0.999**
	Yi-1.5-6B	0.969	0.967	0.991	0.991	0.984	0.919	0.985	0.980	0.951	0.983	0.941	0.982	0.958	0.993	0.982	0.984	0.983	0.959	0.990	0.989	0.963	0.994
	mDeBERTa	0.998	0.998	**0.998**	1.000	**0.996**	0.992	0.991	**0.994**	**0.994**	0.992	0.990	0.998	0.999	**0.999**	0.997	0.997	0.997	0.989	**0.999**	0.999	0.996	0.998
	baseline	0.996	0.996	0.996	1.000	0.990	0.991	0.980	0.991	0.993	**0.997**	**0.992**	**0.999**	**0.999**	0.996	0.997	0.994	0.996	0.992	0.993	**1.000**	0.996	0.991
MultiSocial	Gemma-2-9b-it	0.959	0.985	0.954	0.977	0.957	0.936	0.975	0.969	0.936	0.886	0.811	0.976	0.981	0.958	0.973	0.962	**0.974**	0.968	0.965	0.935	0.952	0.969
	Qwen2-0.5B	0.955	0.985	0.963	0.979	0.953	0.956	0.972	0.967	0.974	0.858	0.822	0.968	0.984	0.956	0.960	0.964	0.950	0.963	0.960	0.881	0.934	0.973
	Qwen2-1.5B	0.967	0.984	0.965	0.976	0.958	0.937	0.977	0.971	0.923	0.857	0.828	0.959	0.956	0.964	0.965	0.967	0.943	0.962	0.960	0.883	0.940	0.979
	Yi-1.5-6B	0.934	0.975	0.956	0.968	0.945	0.931	0.967	0.963	0.966	0.883	0.842	0.960	0.971	0.952	0.958	0.965	0.924	0.951	0.923	0.880	0.929	0.973
	mDeBERTa	**0.973**	0.989	0.943	0.964	0.948	0.961	0.964	0.954	0.971	0.838	0.773	0.957	0.982	0.951	0.957	0.951	0.949	0.956	0.969	**0.941**	0.956	0.977
	baseline	0.967	**0.991**	**0.978**	**0.984**	**0.975**	**0.975**	**0.981**	**0.979**	**0.984**	**0.919**	**0.869**	**0.978**	**0.988**	**0.973**	**0.981**	**0.974**	0.974	**0.970**	**0.971**	0.936	**0.959**	**0.980**

Table 6: Per-language AUC ROC performance of the detectors on in-distribution data. The highest value in each dataset per each language is boldfaced.

		MultiSocial			MULTITuDE_v3			MIX			SemEval			→ Average
Detector	Threshold	FNR	FPR	Macro F1	FNR	FPR	Macro F1	FNR	FPR	Macro F1	FNR	FPR	Macro F1	Macro F1
Gemma-2-9b-it	0.500	0.157	0.052	0.767	**0.026**	0.038	**0.942**	**0.141**	**0.222**	**0.818**	0.128	0.088	**0.891**	**0.855**
Qwen2-0.5B	0.990	0.116	0.071	0.806	0.043	0.076	0.903	0.396	0.374	0.615	0.249	0.133	0.806	0.782
Qwen2-1.5B	0.900	**0.101**	0.109	**0.813**	0.068	0.039	0.877	0.299	0.353	0.674	0.187	**0.038**	0.884	0.812
Yi-1.5-6B	0.900	0.127	0.092	0.788	0.093	0.048	0.841	0.176	0.305	0.758	0.151	0.170	0.840	0.807
mDeBERTa	0.970	0.130	0.043	0.798	0.033	0.022	0.935	0.489	0.296	0.604	0.256	0.133	0.803	0.785
baseline	0.999	0.169	**0.030**	0.761	0.036	**0.018**	0.932	0.520	0.402	0.537	**0.076**	0.375	0.774	0.751

Table 7: Estimation of generic detection performance using the tuned threshold for each detector (for deployment). The best achieved value of each metric is boldfaced (the lowest for FNR and FPR, the highest for Macro F1). The Gemma-2-9b-it detector is offering the best overall detection performance.

differences between baseline and the robustly fine-tuned mDeBERTa-v3-base **are statistically significant**, confirmed by two-sided DeLong test (DeLong et al., 1988) on both MIX and SemEval OOD datasets (p-values < 0.001). The better performance of robustly fine-tuned detectors using our augmented datasets for training clearly **shows the benefits of our proposal**.

The mDeBERTa-v3-base model (the same base model as the baseline) was able to achieve by 21% higher AUC ROC performance on the MIX dataset and by 4% on the SemEval dataset. Regarding the TPR@5%FPR metric, the results show that the baseline detector was not able to achieve such a FPR for any classification threshold. On the other hand, the robustly fine-tuned version was able to detect almost 70% of MGTs on the SemEval dataset for 5% FPR. It must be noted that not only the baseline, but also the Yi-1.5-6B and Qwen2-0.5B detectors have not detected any MGT in the 5% FPR conditions. Overall, the best detector on OOD data is the Gemma-2-9b-it robustly fine-tuned model, detecting 52% and 83% of MGTs of the MIX and SemEval datasets for the 5% FPR conditions.

A deeper analysis using the ROC curves of the detectors for individual datasets (Appendix C) revealed that the used OOD data are really challenging for them, while the in-distribution data are resulting in almost perfect classification capabilities. Figure 5 shows that the performance of the baseline detector is only negligibly better than a random guessing (a diagonal) using the MIX dataset.

5.1 Application in the Real World

Based on such in-distribution and OOD evaluation, we performed a manual classification-threshold tuning to achieve optimal trade-off detection (i.e., the highest Macro F1 across the datasets). **Thresholds are selected to maximize deployment utility rather than to evaluate the proposed robustness-oriented training methodology**, and all robustness claims rest on the AUC ROC threshold-independent metric in the main results above. The tuned thresholds are needed for application of the detectors in the real world (i.e., in the wild, without known labels) to make the decision and to approximate the actual expected performance (e.g., FPR). The thresholds are tuned to balance the performance in each dataset individually and to maximize the performance across all the datasets – i.e., to identify the optimal threshold to be applied in the real world. The results for the tuned thresholds are provided in Table 7, where we provide the FNR, FPR, and Macro F1 metrics for each dataset, as well as the average of Macro F1 across the four datasets. The results indicate the significantly better performance of the Gemma-2-9b-it detector compared to the others, representing also the largest model. The Qwen2-1.5B detector offers also good performance, while being much smaller

than the best model (9B vs 1.5B parameters $\sim 6\times$ smaller); thus, offering a good trade-off between the performance and inference costs.

5.2 Comparison to Existing Detectors

Table 8 summarizes the detection performance results on each of the used datasets achieved by existing best performing statistical detection methods, applicable in zero-shot manner (without training), namely Binoculars (Hans et al., 2024), Fast-DetectGPT (Bao et al., 2023), and LLM-Deviation (Wu and Xiang, 2023). Macro F1 is calculated based on manual threshold tuning, analogously to the results in Table 7. Similarly, Table 9 shows the performance of existing pre-trained detection methods, fine-tuned on different datasets. BLOOMz represents BLOOMZ-3B-mixed-detector (Sivesind and Winje, 2023), Longformer refers to Longformer Detector (Li et al., 2024), and OpenAI represents RoBERTa-Large OpenAI Detector (Solaiman et al., 2019).

The comparison to the existing methods indicates that **even our worst performing fine-tuned detector performs better (based on Macro F1 average) than the usually applied publicly available detectors (statistical or pretrained)**.

Dataset	Metric	Binoculars	Fast-DetectGPT	LLM-Deviation
MultiSocial	**AUC ROC**	**0.7753**	0.7717	0.7454
	TPR@5%FPR	**0.3069**	0.3063	0.1561
	Macro F1	**0.4433**	0.3450	0.3406
MTD_v3	**AUC ROC**	0.8169	**0.8184**	0.6666
	TPR@5%FPR	**0.5647**	0.5576	0.0829
	Macro F1	**0.6510**	0.6328	0.5928
MIX	**AUC ROC**	**0.5369**	0.5147	0.5053
	TPR@5%FPR	**0.2347**	0.2321	0.0928
	Macro F1	**0.5552**	0.5504	0.5089
SemEval	**AUC ROC**	0.8782	0.8835	**0.9098**
	TPR@5%FPR	**0.7353**	0.7087	0.2437
	Macro F1	0.8207	0.8161	**0.8628**
	↓ Average Macro F1	**0.6176**	0.5861	0.5763

Table 8: Evaluation of detection performance of zero-shot statistical detection methods. MTD_v3 denotes MULTITuDE_v3 dataset. The highest value in each row is boldfaced.

Dataset	Metric	BLOOMz	Longformer	OpenAI
MultiSocial	**AUC ROC**	**0.7523**	0.4609	0.6802
	TPR@5%FPR	**0.2233**	0.0377	0.1963
	Macro F1	**0.5715**	0.3924	0.4750
MTD_v3	**AUC ROC**	0.5473	**0.6082**	0.5493
	TPR@5%FPR	0.0914	0.1057	**0.1362**
	Macro F1	0.3654	**0.4862**	0.4753
MIX	**AUC ROC**	0.5625	**0.7274**	0.3997
	TPR@5%FPR	0.0000	**0.2383**	0.0641
	Macro F1	0.5138	**0.6527**	0.4228
SemEval	**AUC ROC**	0.7452	**0.9177**	0.6498
	TPR@5%FPR	0.2967	**0.5663**	0.1616
	Macro F1	0.6189	**0.8059**	0.5611
	↓ Average Macro F1	0.5174	**0.5843**	0.4835

Table 9: Evaluation of detection performance of existing fine-tuned detection methods. MTD_v3 denotes MULTITuDE_v3 dataset. The highest value in each row is boldfaced.

5.3 Adversarial Robustness

For evaluation of adversarial robustness, we use obfuscated MGTs of MULTITuDE_v2 test set in addition to their unobfuscated human samples. These data include obfuscation by 10 authorship obfuscation methods (Macko et al., 2024), covering also methods that have not been used for adversarial training of our detectors. In our evaluation, we use 8 of these 10 obfuscation methods, since the original authors have disqualified DIPPER and Pegasus-paraphrase as unusable in multilingual settings (caused severe damages to the text readability).

The results in Table 10 indicate that **the robustly fine-tuned detectors show strong resilience to these kinds of obfuscations** (AUC ROC values are even slightly higher on obfuscated data than on original data), which otherwise cause evasion of detection. On the other hand, the baseline mDeBERTa-v3-base version has decreased performance in all the cases (up to -27% in case of HomoglyphAttack) except for a slight increase on the ChatGPT paraphrased data. In all the cases, the robustly fine-tuned mDeBERTa achieves higher AUC ROC values than the baseline version.

The MULTITuDE_v2 data are considered in-distribution, since a part of their train set has been combined into the robust training of our detectors. Therefore, a slightly higher performance of mDeBERTa-v3-base in comparison to other robustly fine-tuned detectors is not unexpected (analogously to the performance on MULTITuDE_v3 in Table 4).

Obfuscator	Gemma-2-9b-it	Qwen2-1.5B	mDeBERTa	baseline
original	0.9757	0.9719	**0.9887**	0.9821
adversarial				
ALISON	0.9790	0.9744	**0.9872**	0.9804
DFTFooler	0.9834	0.9783	**0.9946**	0.9776
backtranslation				
m2m100-1.2B	0.9842	0.9840	**0.9918**	0.9780
nllb-200-dist 1.3B	0.9843	0.9878	**0.9913**	0.9819
paraphrasing				
ChatGPT	0.9816	0.9905	**0.9953**	0.9926
text edits				
HomoglyphAttack	0.9860	0.9850	**0.9986**	0.7213
GPTZeroBypasser	0.9880	**1.0000**	0.9987	0.8505
GPTZzzs	0.9806	0.9776	**0.9917**	0.9793

Table 10: Evaluation of adversarial robustness (AUC ROC comparison) of the selected detection methods. The highest value in each row is boldfaced.

6 Ablation Study

To find out the effect of individual components of the proposed robustness-oriented fine-tuning methodology, we executed an ablation study using the mDeBERTa model. For this purpose, we iteratively added individual components, starting with the baseline (v0) and resulting with robustly fine-tuned version (v6) reported in Table 5. For evaluation and comparison of these intermediate versions, we use the OOD datasets (MIX and SemEval) and the AUC ROC metric.

The results are provided in Table 11. It can be seen that balancing the training data using additional human-written texts helps significantly, especially in case of the SemEval dataset. After the inclusion of obfuscated machine-generated data into training, the performance on the MIX dataset is boosted significantly; however, it drops on the SemEval dataset. When switching from in-distribution validation to OOD validation data, the performance is unchanged. HJA modification of a portion of machine-generated data slightly increased the performance on the SemEval data. After using the final text preprocessing (anonymization + lowercasing) of training as well as testing data, v6 achieves peak performance on the MIX dataset, while it is one of the best on the SemEval data. v3 also shows that adding 1% of MIX data (MIX2k) for validation **has not caused an overfit to the test distribution of the MIX dataset**. To rule out the data-distribution leakage effect completely (due to using MIX2k for OOD validation), we trained another detector (v5) by avoiding OOD validation during robust fine-tuning. It thus corresponds to the v2 + HJA + anonymization version. We can see that the performance is very close to the final robust v6 version, significantly better than the baseline in OOD evaluation (also confirmed by two-sided DeLong tests with p-values < 0.001).

To also rule out data size effect (due to training dataset extension), we executed an experiment of using a fixed number of train samples in both, baseline and robust training. The training is stopped after using 1M train samples and the last model checkpoint is selected, i.e., the checkpoint selection is invariant to the validation-set performance. We can observe (the bottom part of Table 11) an increased performance of the baseline on both OOD datasets (indicating the previous overfit to train distribution based on in-distribution validation), while the SemEval performance achieves the top. However, the robust fine-tuning still achieves a significantly better performance than the baseline on the MIX dataset (two-sided DeLong tests with p-values < 0.001). It shows that **the boosted performance of the proposed training methodology is not simply because of more data**, indicating an important role of checkpoint selection during fine-tuning. Also, to simultaneously eliminate the effect of the minority class (highly imbalanced for baseline, with 60k human samples available), we use stricter limits of 100k and 32k train samples. The results show consistently increased performance of robustly trained detector on the MIX dataset, with the difference being statistically significant (DeLong tests showing the highest p-value of 0.0033 in case of 32k fixed train samples, still below the standard 0.05). These findings demonstrate that **class imbalance in the training data is not the primary driver of degraded OOD performance, thereby validating the necessity of the proposed multifactor data diversification strategy**.

Detector	MIX	SemEval
v0 (baseline)	0.5502	0.8305
v1 (baseline + extended data)	0.5824	0.9144
v2 (v1 + obfuscated data)	0.6302	0.8617
v3 (v2 + OOD validation)	0.6336	0.8553
v4 (v3 + HJA)	0.6318	0.8718
v5 (v2 + HJA + anonymization)	0.6497	0.8770
⋆v6 (v4 + anonymization)	**0.6513**	0.8679
Size effect (fixed number of seen train samples)		
v0 (1M)	0.6067	**0.9178**
v6 (1M)	0.6291	0.8281
v0 (100k)	0.6087	0.8248
v6 (100k)	0.6134	0.7072
v0 (32k)	0.6160	0.7121
v6 (32k)	0.6184	0.6266

Table 11: Effect of individual components of robust fine-tuning on detection performance (AUC ROC) on out-of-distribution data. v6 is the robustly fine-tuned mDeBERTa-v3-base (marked by ⋆). The highest value in each column is boldfaced.

7 Conclusions

The fine-tuned MGTD models are known to generalize poorly to out-of-distribution (OOD) data. In this work, we augmented training data to better balance the number of samples of machine and human classes. Besides, we included obfuscated data directly into the detection model training; thus potentially making it more resistant against obfuscation as well. The results showed that such robustly fine-tuned detectors offer lower false positive rates and

the proposed training methodology is able to boost performance (AUC ROC) by up to 21%. OOD robustness in multilingual MGTD depends more on training distribution diversity and OOD-aware model selection than on architecture alone. The performance of different detection models varies, providing an option to select the best suited one giving the trade-off between the detection performance and inference time (or availability of GPU accelerators). The further work is required to optimize the hyper-parameters of the fine-tuning process to boost the detection performance of the best detection models even higher while offering even better generalizability to OOD data. Future work could also explore learned or adaptive obfuscation-detection strategies that generalize beyond the specific perturbation patterns used in this work.

Limitations

Although we have tried to cover variety of domains, generators, and languages, the study is still limited in out-of-distribution evaluation. There is much more space to explore, and new text generation LLMs appear each month that could cause further performance degradation. Our work improves robustness to currently known distribution shifts, but it does not solve generalization against future unseen generators permanently. We have also done a manual hyper-parameters optimization of fine-tuning process; however, we have covered just a small set of options, where a further tuning might increase the generalizability of the detectors.

Ethical Considerations

Our work is focused on making the detection methods more robust to out-of-distribution data. We use the existing datasets in our work in accordance with their intended use and licenses (for research purpose only). As a part of our work, we are not re-sharing any existing data or publishing any new dataset. For the research replicability and validation purposes, we are publishing the pre-processing, training, and evaluation source codes (for research purposes only). We have not used AI assistants in our work except for text polishing.

Acknowledgments

This work was partially supported by the European Union under the Horizon Europe project AI-CODE, GA No. 101135437; by the EU NextGenerationEU through the Recovery and Resilience Plan for Slovakia under the project No. 09I01-03-V04-00059; and by the Slovak Research and Development Agency under the project Modermed, GA No. APVV-22-0414. We acknowledge EuroHPC Joint Undertaking for awarding us access to Leonardo at CINECA, Italy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.AI. *Preprint*, arXiv:2403.04652.

Wissam Antoun, Virginie Mouilleron, Benoît Sagot, and Djamé Seddah. 2023. Towards a robust detection of language model-generated text: Is ChatGPT that easy to detect? In *Actes de CORIA-TALN 2023. Actes de la 30e Conférence sur le Traitement Automatique des Langues Naturelles (TALN), volume 1 : travaux de recherche originaux – articles longs*, pages 14–27, Paris, France. ATALA.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2023. Fast-DetectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Ali Borji. 2023. A categorical archive of chatgpt failures. *Preprint*, arXiv:2302.03494.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Yiran Chen, Zhenqiao Song, Xianze Wu, Danqing Wang, Jingjing Xu, Jiaze Chen, Hao Zhou, and Lei Li. 2022. MTG: A benchmark suite for multilingual text generation. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2508–2527, Seattle, United States. Association for Computational Linguistics.

Evan N Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*, 11:70977–71002.

Elizabeth R DeLong, David M DeLong, and Daniel L Clarke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. *Biometrics*, pages 837–845.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, 36.

Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. 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)*, pages 12463–12492, Bangkok, Thailand. Association for Computational Linguistics.

Tiziano Fagni, Fabrizio Falchi, Margherita Gambini, Antonio Martella, and Maurizio Tesconi. 2021. TweepFake: About detecting deepfake tweets. *Plos one*, 16(5):e0251415.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Saadia Gabriel, Liang Lyu, James Siderius, Marzyeh Ghassemi, Jacob Andreas, and Asu Ozdaglar. 2024. Generative ai in the era of'alternative facts'.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: zero-shot detection of machine-generated text. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2022. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

Xinlei He, Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2024. MGTBench: Benchmarking machine-generated text detection. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2251–2265.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2023. RADAR: Robust AI-text detection via adversarial learning. *Advances in Neural Information Processing Systems*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. 2024. OUTFOX: Llm-generated essay detection through in-context learning with adversarially generated examples. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21258–21266.

Mina Lee, Percy Liang, and Qian Yang. 2022. Coauthor: Designing a human-ai collaborative writing dataset for exploring language model capabilities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Yafu Li, Qintong Li, Leyang Cui, Wei Bi, Zhilin Wang, Longyue Wang, Linyi Yang, Shuming Shi, and Yue Zhang. 2024. MAGE: Machine-generated text detection in the wild. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–53, Bangkok, Thailand. Association for Computational Linguistics.

Lang Liu, Krishna Pillutla, Sean Welleck, Sewoong Oh, Yejin Choi, and Zaid Harchaoui. 2021. Divergence frontiers for generative models: Sample complexity, quantization effects, and frontier integrals. *Advances in Neural Information Processing Systems*, 34:12930–12942.

Jason Lucas, Adaku Uchendu, Michiharu Yamashita, Jooyoung Lee, Shaurya Rohatgi, and Dongwon Lee. 2023. Fighting fire with fire: The dual role of LLMs in crafting and detecting elusive disinformation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14279–14305, Singapore. Association for Computational Linguistics.

Dominik Macko, Jakub Kopál, Robert Moro, and Ivan Srba. 2025. 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)*, pages 727–752, Vienna, Austria. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Jason Lucas, Michiharu Yamashita, Matúš Pikuliak, Ivan Srba, Thai Le, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2023. MULTITuDE: Large-scale multilingual machine-generated text detection benchmark. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9960–9987, Singapore. Association for Computational Linguistics.

Dominik Macko, Robert Moro, Adaku Uchendu, Ivan Srba, Jason S Lucas, Michiharu Yamashita, Nafis Irtiza Tripto, Dongwon Lee, Jakub Simko, and Maria Bielikova. 2024. Authorship obfuscation in multilingual machine-generated text detection. In *Findings*

of the Association for Computational Linguistics: EMNLP 2024, pages 6348–6368, Miami, Florida, USA. Association for Computational Linguistics.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. 2021. MAUVE: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34:4816–4828.

Jiameng Pu, Zain Sarwar, Sifat Muhammad Abdullah, Abdullah Rehman, Yoonjin Kim, Parantapa Bhattacharya, Mobin Javed, and Bimal Viswanath. 2023. Deepfake text detection: Limitations and opportunities. In *2023 IEEE symposium on security and privacy (SP)*, pages 1613–1630. IEEE.

Jiashu Pu, Ziyi Huang, Yadong Xi, Guandan Chen, Weijie Chen, and Rongsheng Zhang. 2022. Unraveling the mystery of artifacts in machine generated text. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6889–6898, Marseille, France. European Language Resources Association.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Domenic Rosati. 2022. SynSciPass: detecting appropriate uses of scientific text generation. In *Proceedings of the Third Workshop on Scholarly Document Processing*, pages 214–222, Gyeongju, Republic of Korea. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2025. Can AI-generated text be reliably detected? stress testing AI text detectors under various attacks. *Transactions on Machine Learning Research*.

Areg Mikael Sarvazyan, José Ángel González, Marc Franco-Salvador, Francisco Rangel, Berta Chulvi, and Paolo Rosso. 2023. Overview of AuTexTification at IberLEF 2023: Detection and attribution of machine-generated text in multiple domains. *arXiv preprint arXiv:2309.11285*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Tatiana Shamardina, Vladislav Mikhailov, Daniil Chernianskii, Alena Fenogenova, Marat Saidov, Anastasiya Valeeva, Tatiana Shavrina, Ivan Smurov, Elena Tutubalina, and Ekaterina Artemova. 2022. Findings of the the RuATD shared task 2022 on artificial text detection in russian. In *Computational Linguistics and Intellectual Technologies*, page 497–511. RSUH.

Nicolai Thorer Sivesind and Andreas Bentzen Winje. 2023. Machine-generated text-detection by fine-tuning of language models.

Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, Gretchen Krueger, Jong Wook Kim, Sarah Kreps, et al. 2019. Release strategies and the social impacts of language models. *arXiv preprint arXiv:1908.09203*.

Reuben Tan, Bryan Plummer, and Kate Saenko. 2020. Detecting cross-modal inconsistency to defend against neural fake news. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2081–2106, Online. Association for Computational Linguistics.

Gemma Team. 2024. Gemma.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pawan Kumar Verma, Prateek Agrawal, Ivone Amorim, and Radu Prodan. 2021. WELFake: Word embedding over linguistic features for fake news detection. *IEEE Transactions on Computational Social Systems*, 8(4):881–893.

Ivan Vykopal, Matúš Pikuliak, Ivan Srba, Robert Moro, Dominik Macko, and Maria Bielikova. 2024. Disinformation capabilities of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14830–14847, Bangkok, Thailand. Association for Computational Linguistics.

Jan Philip Wahle, Terry Ruas, Frederic Kirstein, and Bela Gipp. 2022. How large language models are transforming machine-paraphrase plagiarism. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 952–963, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Osama Mohammed Afzal, Tarek Mahmoud, Giovanni Puccetti, and Thomas Arnold. 2024. SemEval-2024 task 8: Multidomain, multimodel and multilingual machine-generated text detection. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 2057–2079, Mexico City, Mexico. Association for Computational Linguistics.

Zhendong Wu and Hui Xiang. 2023. MFD: Multi-feature detection of LLM-generated text. *PREPRINT (Version 1) available at Research Square*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. *Advances in neural information processing systems*, 32.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. 2023. Red teaming ChatGPT via jailbreaking: Bias, robustness, reliability and toxicity. *Preprint*, arXiv:2301.12867.

Aneta Zugecova, Dominik Macko, Ivan Srba, Robert Moro, Jakub Kopál, Katarína Marcinčinová, and Matúš Mesarčík. 2025. Evaluation of LLM vulnerabilities to being misused for personalized disinformation generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 780–797, Vienna, Austria. Association for Computational Linguistics.

A Computational Resources

For experiments regarding model fine-tuning and inference processes, we have used 1x A100 64GB GPU, cumulatively consuming around 1,000 GPU-core hours. For data augmentation, pre-processing, combining, and analysis of the results, we have used Jupyter Lab running on 4 CPU cores, without the GPU acceleration.

Regarding the environment, we have used Python 3.11.6 with key libraries of pytorch 2.6.0 (with CUDA 11.8), transformers 4.57.6, accelerate 1.12.0, bitsandbytes 0.48.2, and peft 0.18.1.

B MIX Dataset

For the diverse mixture dataset of existing labeled datasets for out-of-distribution evaluation, we have combined the samples of data (up to 5,000 text samples per language/source and label) of the following datasets: TuringBench (Uchendu et al., 2021), Grover (Zellers et al., 2019), GPT-2 (Radford et al., 2019), GPT-3 (Brown et al., 2020), Paired (Pu et al., 2022), CNN/DailyMail (See et al., 2017), NeuralNews (Tan et al., 2020), InTheWild (Pu et al., 2023), Wikitext (Merity et al., 2016), WritingPrompts (Fan et al., 2018), CoAuthor (Lee et al., 2022), TweepFake (Fagni et al., 2021), WELFake (Verma et al., 2021), AuTexTification (Sarvazyan et al., 2023), MTG (Chen et al., 2022), MGTBench (He et al., 2024), RuATD (Shamardina et al., 2022), SynSciPass (Rosati, 2022).

The data have been combined in a common way, providing texts, labels, source, and if available also the language and domain/topic information. Afterwards, the duplicates of the texts have been removed and the final dataset has been pseudo-randomly sampled to include 100k samples for the human and machine classes. For the MIX2k validation dataset, the 1k samples of each class have been pseudo-randomly sampled. It resulted into diverse data of at least (since some are unspecified in the original datasets) 7 languages (cn, de, en, es, fr, ru, zh), 10 domains (news, tweets, unspecified social-media texts, comments, reviews, wiki texts, stories, literature, scientific articles, question answering, commonsense reasoning), 50 generators (sizes of under 1B parameters to over 100B, including pure generators, but also translators, co-writing agents, as well as unknown reddit bots), and various generation settings (e.g., diverse approaches, at least 7 different sampling strategies).

Due to various licensing of the data of the used datasets (some of which are also composites of multiple data sources), we are not re-sharing the dataset itself. However, for the research purposes (validation and replication), we are providing the data construction source code.

C Supplemental Results

In Figure 3–6, ROC curves of the detectors for individual datasets are provided.

In Figure 7 and 8, we provide a deeper analysis of the performance of the best detector of Gemma-2-9b-it on the most challenging MIX dataset. The sample counts are not balanced across the provided categories; therefore, the results are just indicative. There is a significant drop in performance for German and French languages, which are included only for machine-generated samples originated from machine translation (MTG dataset). The low performance is also observed in the commonsense-reasoning, wikitexts, and literature domains.

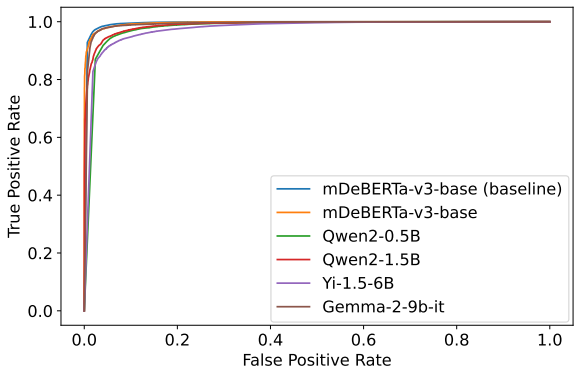


Figure 3: Receiver operating characteristic curves of the detectors using the MULTITuDE_v3 dataset.

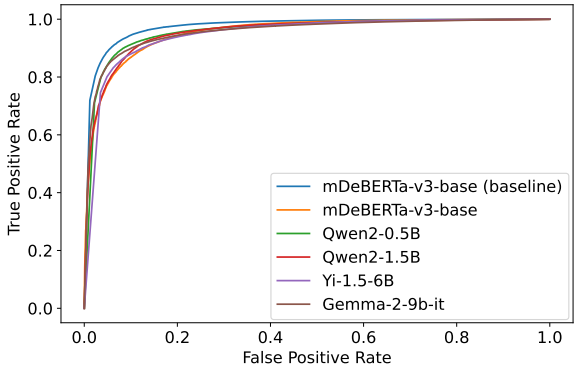


Figure 4: Receiver operating characteristic curves of the detectors using the MultiSocial dataset.

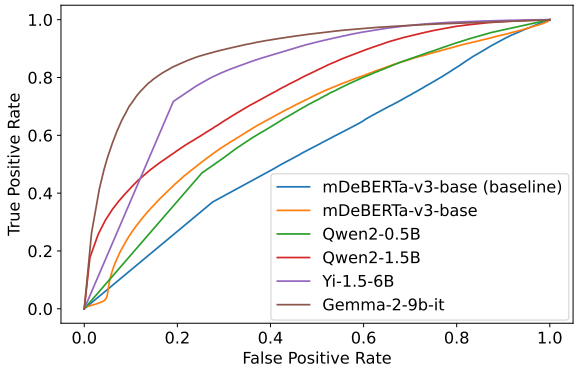


Figure 5: Receiver operating characteristic curves of the detectors using the MIX dataset.

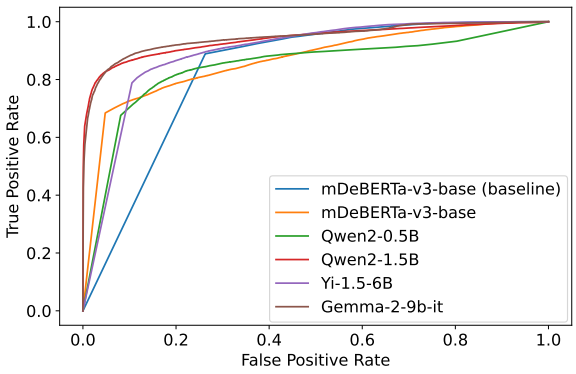


Figure 6: Receiver operating characteristic curves of the detectors using the SemEval dataset.

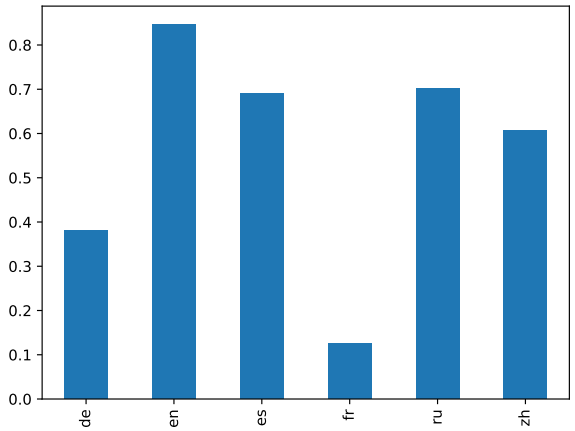


Figure 7: Per-language performance (Macro F1) of the best Gemma-2-9b-it detector.

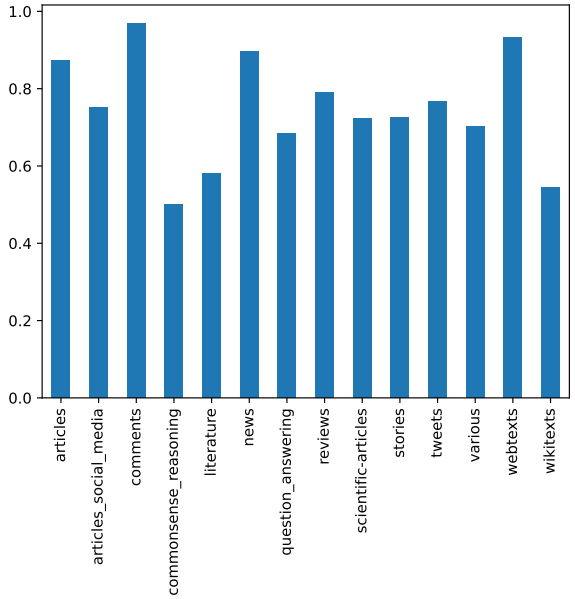


Figure 8: Per-domain performance (Macro F1) of the best Gemma-2-9b-it detector.

LLMs are Easily Startled: Fragility Testing with Lookalike Character Noise

Peter deJonge[1] **and Nathanael Chambers**[1,2]
[1]US Naval Academy, Annapolis MD
[2]Williams College, Williamstown MA
nc19@williams.edu

Abstract

Language models are dependent on the character sets of the languages seen during training. Some languages use large character sets, like Chinese and Korean, while others are very small, like English. When the input during inference diverges from the trained character set, it can lead to unexpected behavior. While many researchers have looked at using this weakness for adversarial attacks, these studies typically focus on extreme cases that lead to model failure. But we don't have a strong understanding of how brittle models are to more casual and routine character substitutions. In fact, many 'known' characters have Unicode lookalikes that users enjoy using without malicious purposes. This paper studies the *fragility* of models when faced with such character noise. When input contains innocuous character substitutions, do models robustly handle the distortion? How does the tokenizer and overall model size affect performance? This paper presents analyses of tokenizers and a series of fragility tests to answer these questions. We find that tokenization is not the only stumbling block, and neither is a larger model size the assured solution. We present a formal benchmark, publicly release a Fragility Scorer[1], and describe our experimental results on a variety of open-source models. Implications and suggestions for improving fragility are then presented.

1 Introduction

Language models are trained on sequences of tokenized text, and a wealth of research has shown that input which contains unexpected formatting that diverges from training formats can lead to strange behavior. One particular kind of formatting is lookalike character substitution (ρ -> p). These are used in a lighthearted manner on social media, or for more intentional information hiding purposes that humans easily read, but which fool automated extractors. This paper studies what we call the "fragility" of models when facing general character distortion and manipulation. In particular, we look at how model size and its tokenizers affect robustness.

A number of research efforts have focused on manipulating model behavior by altering its input. These include jailbreaking (Yi et al., 2024), adversarial prompting (Chao et al., 2025), synonym swapping (Li et al., 2024), gradient-based attacks (Ren et al., 2025), and many others. Most of these focus on an intentional goal of altering system behavior, thus proposing methods to discover inputs to produce maximum impact. This paper, however, broadens the aperture of study to the *general* performance of models in the face of character manipulation. While a model might fail on a specific attack, it could still be robust to 99% of manipulations that occur in typical digital writing.

The goals of this work are partially motivated by recent findings that large language models (LLMs) struggle on simple character and word-level challenges. Efrat et al. (2023) created a series of tests for word manipulation (e.g., 'what word occurs after "the" in the following sentence') that illuminated the struggle between model tokenization and lexical manipulation that humans perform with ease. Despite the many strengths of LLMs on complex tasks, they struggle on 'easy' ones. We thus look deeper into character-level changes and how they might affect downstream complex tasks.

Several societal challenges also motivate the importance of understanding model fragility. We take inspiration from human trafficking prevention where Unicode character swapping is observed in illegal online advertisements. Phone numbers are sometimes masked with digit lookalikes (e.g., ∅ not 0). Again, the malicious actor isn't trying to break a model per se, but instead just hopes to hide the digits. Can LLMs still extract informa-

[1]https://www.github.com/nchambers/fragility

Proceedings of the 11th Workshop on Natural User-generated Text, pages 167–181
October 28, 2026 ©2026 Association for Computational Linguistics

tion from text that diverges from its trained character set? How fragile are today's models? Does model size correlate with fragility? We investigate these questions through a series of fragility tests.

The main contributions of this work include: (1) A carefully chosen set of character swapping tests to evaluate model fragility, (2) Fragility scores from 7 families of open-source models across 24 size variants, (3) Results from a variety of prompting approaches, from few-shot to the inclusion of unicode mappings (e.g., telling the model that α is a), (4) The first analysis of how Unicode substitutes are represented by tokenizers in comparison to their ASCII lookalikes, (5) A task for phone number extraction motivated by the domain of human trafficking, and (6) Release of the Fragility Scorer software to test future models.

In summary, this paper provides analyses on how Unicode lookalikes are represented in Transformer-based models, how noise affects performance, how model size affects fragility, and how today's models compare to each other.

2 Related Work

Research into adversarial attacks on LLMs is related to this paper, but tends to focus more on inputs that achieve specific behavior. Jailbreaking manipulates the prompt to circumvent model instructions. A variety of helpful survey papers are available (Knowlton et al., 2026; Yi et al., 2024; Tian et al., 2025; Das et al., 2025). Gradient-based attacks access model parameters to discover input tokens that most affect the final distribution (Li et al., 2018, 2020; Zou et al., 2023; Ren et al., 2025), crafting attacks that alter model output. Many attacks add instructions to the prompt (Liu et al., 2024; Chao et al., 2025), some using multilingual substitutions (Deng et al., 2024). One approach that is a bit closer to this paper is synonym swapping: searching for input tokens to replace for maximum effect (Li et al., 2024; Wang et al., 2024; Zheng et al., 2025b). A first look at homoglyph attacks on encoders came from Boucher et al. (2022). We differ from the bulk of attack-based research in that we are not trying to break systems nor are we trying to find the *best* attack, but rather we seek to measure overall fragility of models when facing general character manipulation, some of which is innocuous and unintentional.

Most relevant to this paper is prior work on character substitutions. There is a nice thread of recent work on how traditional *classifiers* (not LLM generation) are affected by Unicode or other character substitutions. Edman et al. (2024) proposed a series of character *tasks* for LLMs to perform, such as swapping letters and reversing words. CharBench is a similar set of tasks for character analysis (Uzan and Pinter, 2026). Yuan et al. (2024) study character cipher substitutions. A number of people have looked at LLM awareness of substrings (Ciaccio et al., 2025; Uzan and Pinter, 2026). Some have studied tokenization and how breaking up tokens in English (Zheng et al., 2025a) and Chinese (Haslett, 2025) affects LLM performance. Zhang et al. (2021) was one of the first studies of how BERT-based classifiers perform when training/testing with character substitutions. They found that performance was poor, but could be improved with training data augmentation. This is obviously not an option for today's largest models without direct access to computational resources beyond most researchers.

Finally, work on visual models proposes new architectures that use CNNs to build the input characters' visual embeddings (Liu et al., 2017; He et al., 2018; Summers et al., 2023), thus learning to identify Unicode and ASCII similarity more like a human's eye (Eger et al., 2019; Salesky et al., 2021; Rust et al., 2023). The idea is not to use character (or token) embeddings as the input, but visual images that are transformed by CNNs. While an interesting research area, this paper does not focus on CNN architectures or building custom LLM frameworks, but instead on the standard Transformer of today's most used models.

3 Fragility Tests

This section describes our series of fragility tests to stress-test language models for robustness in character noise. We separate these into three types of tests: letters, digits, and broader language understanding. The digit tests are grounded in the extrinsic task of phone number identification. The letter tests are grounded in word identification, and understanding uses a common Q&A benchmark that tests language understanding ability.

3.1 Lookalike Dataset

For all experiments in this paper, we use the open-source homoglyph dataset[2]. Specifically, we use its raw character clusters that represent lookalike

[2]www.github.com/codebox/homoglyph

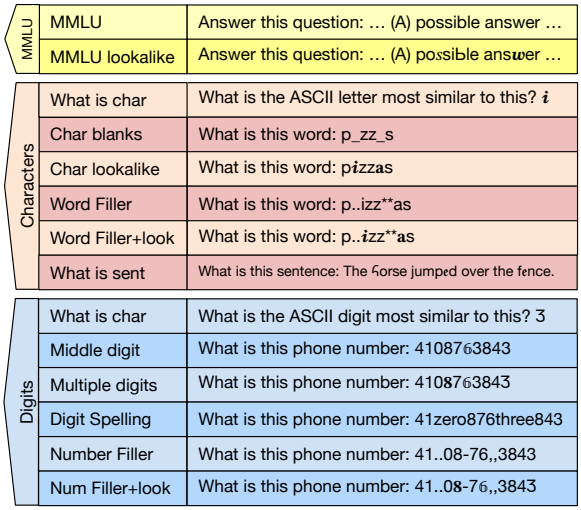


Figure 1: The subtasks in the Fragility Score.

Unicode characters based on the ASCII digits 0-9 and letters a-z and A-Z. Its creators constructed this dataset by manually identifying all of the lookalikes they could find. We did not augment or alter the dataset, protecting from any bias from this paper's authors. An example of the q character lookalikes is shown here:

ԛqqԛ ԛ 𝗊𝑞𝑞𝑞𝑞ԛ𝗊ԛ𝗊𝗊𝑞𝑞𝗊

3.2 Baseline Character Identification

We start with a baseline fragility test that asks in the most straightforward manner, *what is this character?* We query the LLM to identify which ASCII character is most similar (and visual) to the given Unicode character. We include two separate baseline tests that instead specifically ask for a digit or a letter, as appropriate. For instance:

```
What ASCII digit is most similar
to this unicode character: 3
```

The correct answer is obviously 3. We provide three examples as a few-shot prompt so smaller models (and models without instruction tuning) have their best chance to demonstrate its learned character set knowledge. See Appendix A.1 for more prompts.

3.3 Letter Understanding

Identify the Word with Blanks: This baseline test evaluates a model's ability to "fill in the blanks" when given a word with missing letters.

p_zz_s → pizzas

This basic test sets the stage for the next evaluation that inserts Unicode lookalikes into the blanks. A model's performance on this blank test measures a model's ability to identify a partial word with no extra information.

We selected 300 English words of length 6 at random. Two of the characters are randomly removed and replaced with underscore characters. This word length was chosen to balance rarity (longer words are easier to guess) with having a good mix of visible and hidden characters.

Identify the Word with Lookalikes: This test uses the same words as above, but instead of underscore characters, Unicode lookalikes are substituted into the underscore locations:

pizzas — pizzas

If a model has a learned representation of lookalike character usage, it should perform better on this test than the prior underscore test.

Identify the Word with Noise: This test uses the same 300 words in the above tests, but with all the original ASCII. The test then inserts punctuation between randomly selected characters in the words, testing a model's ability to understand a word in the presence of confounding noise.

We created two versions of this test. One plain test with just ASCII and noise, and the other uses the Lookalikes test above with added noise. The former is called *Word Noise*, and the latter *Noise+Look*.

Identify the Sentence with Lookalikes: While the word tests above give the most straightforward insight into model understanding of letters in context, the sentence-level test gives the model *more* context by allowing the surrounding words to assist the LLM. This test is given a sentence with lookalike character swaps, and the model must repeat the sentence in ASCII form. To provide the optimal environment for success, we also provide three few-shot examples in the prompt. One example sentence is here:

The ϛorse jumped over the fence.

A correct answer is a generation of the sentence (case-insensitive) with only ASCII characters. The scorer does not penalize for extra whitespace.

3.4 Digit Understanding

The fragility tests for digit recognition are grounded in a real-world task, rather than generic digit prompts. We chose the task of phone number extraction, motivated by its applicability to

downstream applications. One noteworthy application area is human trafficking prevention where phone numbers are the strongest indicator for linking online advertisements (Hundman et al., 2018; Rodríguez Díaz, 2021; Vajiac et al., 2023; Saxena et al., 2023). Phone numbers also serve well as an evaluation because LLMs are thus primed to generate 10 digits.

Number Recognition: Middle Digit This first test provides a phone number where the middle digit has been replaced with a Unicode lookalike character. We prompt the LLM to generate the original phone number from this perturbed input in its entirety. This allows us to evaluate not just that the middle digit is recovered, but also that the model does not confuse the generation of the surrounding ASCII digits either.

Number Recognition: Multiple Digits This test randomly substitutes 33% of the digits with a lookalike character. We run several versions of this test with increasing percentages.

Number Recognition: Noise This test evaluates the fragility of models when the input is not 'properly' formatted, but instead contains extra whitespace and punctuation between characters, such as in '*4=1..0-93,485 31*'.

Number Recognition: Semantic Substitution This test replaces digits with spelled out versions (3 → three), including a random choice of lowercase and uppercase characters (3 → tHReE). Different from the prior tests, only ASCII characters are used, but this test measures the fragility of models when forced to sequentially understand character sequences as digits (four5one vs 451).

Combinations Finally, the fragility tests can create combinations of the above, such as a test that includes both Multiple Digits and Noise. We present results on this at different percent likelihoods in the data.

3.5 Language Understanding

While the above tests evaluate word-level handling, they don't evaluate broader understanding and they don't directly measure how semantic understanding is affected by character noise.

Our goal is to measure model performance on common Q&A benchmarks before and after lookalike substitutions are made. However we do not perturb the benchmark text in its entirety, but rather keep the questions untouched, allowing the model to fully understand the question. This fragility evaluation only alters the multiple-choice *answers*, keeping the question clear. We test a model's ability to extract information from it. See Figure 2 for an example.

```
The surface of Planet Earth loses
energy to outer space due mostly to
 A. conduction.
 B. convection.
 C. radiation.
 D. radioactivity.
```

Figure 2: Example MMLU question whose answers contain Unicode noise. The question is unchanged.

We evaluate fragility of Q&A using the 57 subject MMLU benchmark (Hendrycks et al., 2021). For each LLM, we measure its current accuracy without text changes, and then again but with 20% of the characters in the answers swapped with lookalike substitutes. The score is computed as $(1 - acc20/acc)$. If a model struggles to generate the answer in an extractable format, it is marked incorrect, but the ratio only penalizes on the noise's effect. See Appendix A.5 for more details.

3.6 Chain of Dictionary

Recent work in machine translation experimented with the idea of prompting models with translation equivalents (Lu et al., 2024). For instance, insert 'casa means house' into the prompt. We adopt this idea, called *chain of dictionary* (CoD), for lookalikes with the homoglyph replacements in the prompt: 'I is 1' if the character 'I' appears in the phone number. The digit tests in Figure 1 are duplicated, but with CoD hints, and we separately record their CoD accuracy scores.

4 Tokenizer Analysis

Before the input to an LLM is transformed through its neural layers, the immediate effect of noisy input is observed in how the model's tokenizer splits the text. When unknown characters are in the input, a sentence that might normally be tokenized into a few tokens can expand into 20+ tokens even if just a few Unicode characters are present. An example with Gemma's tokenizer is shown in Figure 3.

Our initial experiments are a set of exploratory tokenization and token embedding measurements to develop an assessment of each tokenizer that

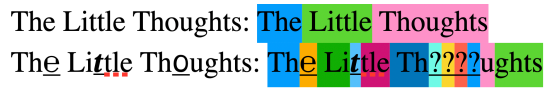


Figure 3: Gemma's tokenizer: altering 3 characters results in 12 tokens vice 3. Question marks show where the 4 bytes in unicode 'o' are tokenized separately.

can then be compared to actual fragility performance. A tokenizer that requires more tokens than another is not necessarily worse off; its network layers could recover meaning from a fragmented input. This analysis measures the correlation between tokenization and performance.

We analyze tokenizers from Llama (also Vicuna), Qwen (also Deepseek), Mistral, Gemma, & OLMo. Our focus is along three dimensions:

- **Token recognition**: what percent of Unicode lookalikes appear in the tokenizer?
- **Embedding similarity**: are Unicode embeddings (as tokens) similar to their nearest-neighbor ASCII characters?
- **Fertility**: how does tokenization increase with character substitutions?

For **token recognition**, we counted how many Unicode lookalikes appear in the tokenizer as standalone tokens or with a whitespace prefix. Figure 4 shows the percent found in the tokenizer (blue bars). There is a clear and noticeable difference in tokenization, with Gemma and Qwen (Deepseek uses Qwen) representing 50% of them while the others are less than 10%. We then measured **embedding similarity** of these tokens to see if their embeddings capture ASCII similarity (i.e., is the Unicode character similar to its visual ASCII equivalent?). We find the Top 3 most similar other tokens using cosine similarity, and see if the ASCII lookalike is among them. The orange bar shows that Gemma and Qwen have learned this similarity, but interestingly, Deepseek's further training of Qwen drastically reduced it.

Finally, we measured the **fertility** of tokenizers that experience tokenization explosion when an input contains lookalikes (Figure 4 shows how one example affects Gemma). This experiment took 2000 examples from the MMLU benchmark, and tokenized them with and without 20% of the characters replaced with Unicode lookalikes. We compute the average number of tokens with and without. Figure 5 shows the averages from each tokenizer. The dark shaded interior shows that they

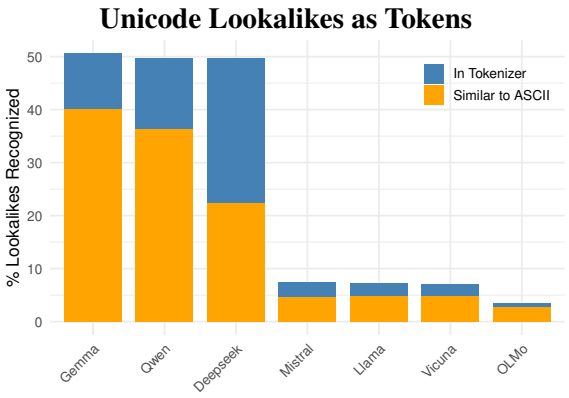


Figure 4: Percent of unicode lookalikes that appear in tokenizers as a standalone token (blue+orange). The orange bar indicates how many known tokens have an embedding similar to its ASCII lookalike. Qwen and Deepseek(Qwen-based) use the same tokenizer, but their embeddings are different.

Model	Sizes Tested
Gemma 3	1B, 4B, 12B, 27B
Llama 3.2	1B, 3B, 70B
OLMo 2	1B, 7B, 13B, 32B
Vicuna 1.5	7B, 13B
Deepseek R1-Qwen	1.5B, 7B, 14B, 32B
Qwen 3	.6B, 1.7B, 4B, 8B, 14B
Mistral-Small	23B
Mixtral-8x7B	7B/56B
GPT-4.1, 5.4, 5.5, 4.1mini, 4omini, 5.4mini, 5mini	unknown

Table 1: Models scored by the Fragility Benchmark.

are similar on regular ASCII text, but lookalike noise causes varying amounts of tokenization explosion. Vicuna and Mistral experience the most drastic expansion requiring 250% more tokens.

We discuss this tokenizer analysis in context of the fragility results after the next section.

5 Experimental Setup

5.1 Models

Table 1 lists the open-source models and sizes used in our experiments. We include both small and medium-sized models to assess how size may or may not affect a Transformer architecture's ability to handle character substitutions.

We also include several models from the OpenAI suite to provide a general comparison against mini and large closed-source models. Other closed LLMs exist, but this work is not an engineering experiment to stress-test black box models. Regrettably, the labs and corporations behind these models have not recently released their latest model ar-

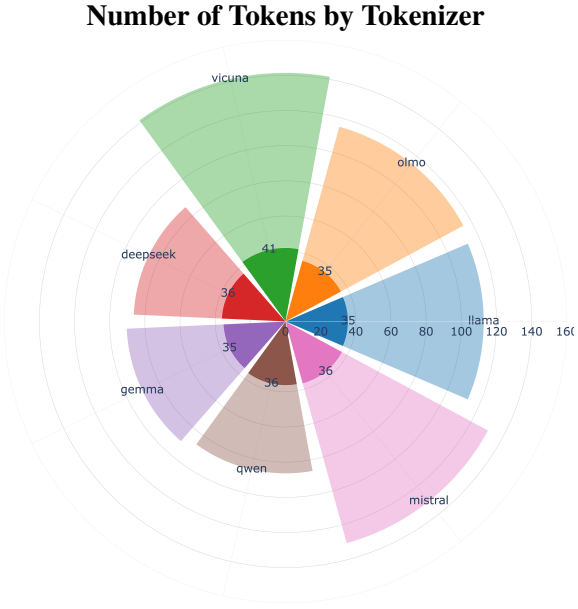


Figure 5: Visualization of how many tokens are produced on normal text (dark shaded) and when 20% is replaced with Unicode lookalikes (lighter). Vicuna and Mistral produce over 250% more tokens.

chitecture changes, training data, nor training procedures. Since few meaningful conclusions from their performance can be drawn without an open release of their architecture, we do not expend computational resources beyond these models. As will be seen, flagship closed-source models also struggle with fragility.

5.2 Fragility Tests and Prompting

We prompt each model (in Table 1) for each test within the Fragility Score (in Section 3) and report individual results for the main tests, as well as one aggregate fragility score.

The prompt templates for each test were chosen based on author experimentation using general instruction language. We altered the prompts with more instructions until generation was reliable in format across the models. Most importantly, we created these prompts using several different models during initial development. The final prompts used for each test are provided in appendix A.1.

The Fragility Score (FS) is a linear combination of accuracies over the three subsets of test types:

$$FS = \lambda_L Letter + \lambda_D Digit + \lambda_U NLU \quad (1)$$

where the lambda weights sum to one and the sub-scores are themselves linear combinations of individual test accuracies (each accuracy ranges $[0, 100]$). We report with $\lambda_L = 0.3$, $\lambda_D = 0.3$, $\lambda_U = 0.4$, but this formulation allows for adjusting the weights to any task balance.

6 Results

All model Fragility Scores can be seen in Figures 6 and 7. At a high-level, there is a trend of an improved fragility score as model size increases, but it is not consistent across all model types. There is a noticeable cluster of models that have fewer parameters yet display much more robustness in the fragility score than the others (circled in the scatterplot). Gemma and Qwen stand out as more robust starting at the ~4B model size. Qwen has a straight line of performers no matter the model size, as does Gemma. You can also see 14B Deepseek in the circle, and its larger 32B model doesn't improve further. Model size is clearly not the sole factor determining fragility. See Appendix A.3 for establishing confidence in these scores.

Figure 6 provides more insight into the different fragility components. We see the Qwen family as the strongest set of models on overall Fragility. Its small 4B model shines amidst the other top performers with larger 14-32b sizes. Gemma is not far behind. But not all models are consistent across the subparts. Several are fragile on digits despite being more robust on letters. The OLMo 32b model is one of the bigger models, and arguably the best on NLU/text, but it struggles as an outlier on digit handling.

Are lookalike characters always disruptive? We compared the two 'what is this word' tests (underscores versus Unicode) to better understand how the model makes its predictions. Interestingly, the Unicode results are substantially higher than the underscore results across all of the models, often with 2x the accuracy. This illustrates how lookalikes are not purely noise, and are more informative than blanks. Appendix A.4 provides more details. We also analyze the effects of punctuation noise in appendix A.6. The median change in phone number prediction from 20% to 80% added noise is a 47% accuracy drop.

How is NLU affected? The MMLU benchmark subscore is with 20% character replacement, but we also tested 40% and 60% in Figure 8. Mistral and Vicuna see the largest drops, effectively breaking their NLU capability. The best tokenizing models do better, but it's not consistent. Qwen drops almost 50% on NLU, despite it outperforming on other tasks. It's clear that tokenization is

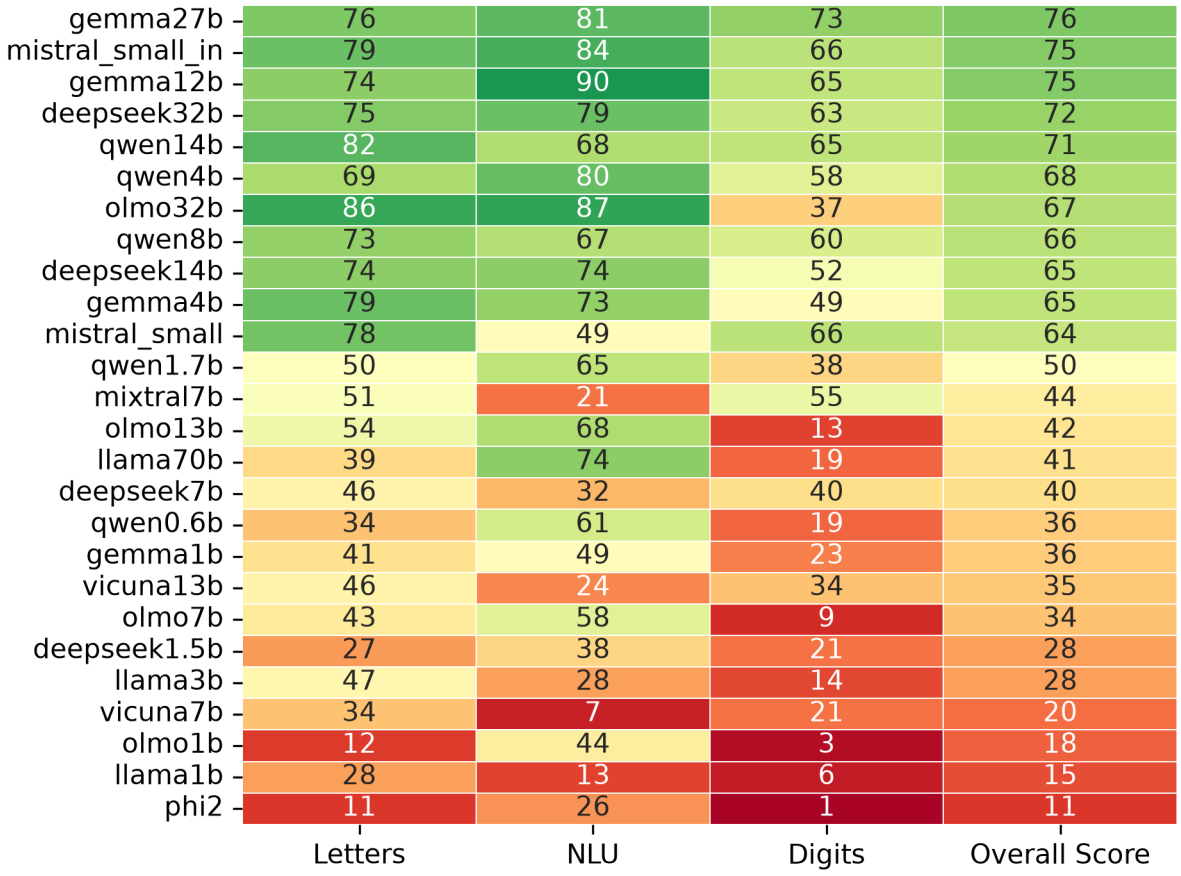


Figure 6: Model scores on individual components of the Fragility Score. [0-1] shown as [0-100] for readability.

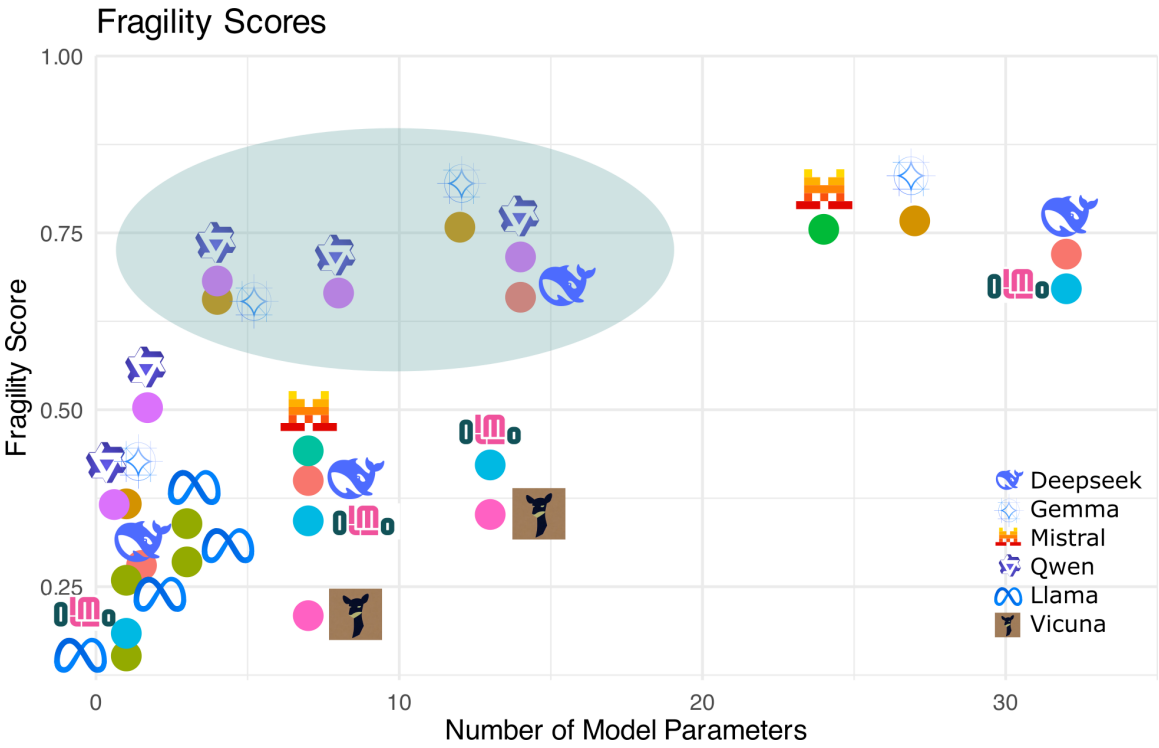


Figure 7: Fragility score for each model, comparing model size (in billions) to its score. The circle indicates outliers that outperform despite having fewer parameters. Qwen and Gemma are less fragile even in smaller sizes.

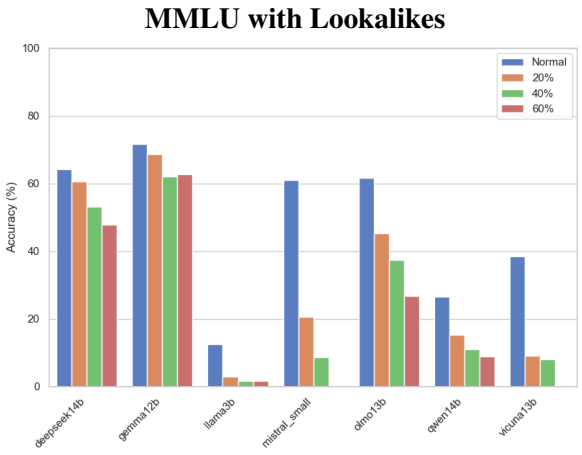


Figure 8: MMLU accuracy at different amounts of characters in the answers replaced with Unicode lookalikes.

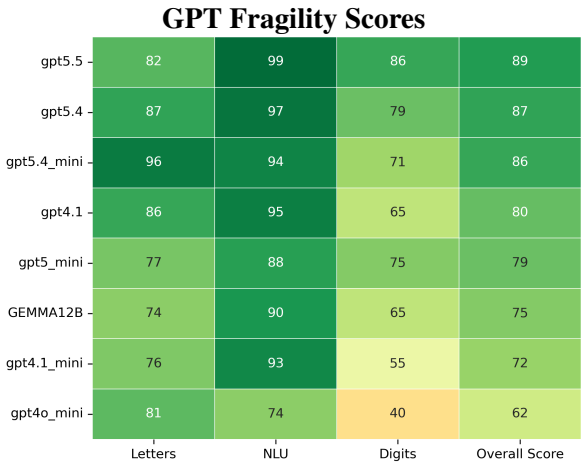


Figure 9: GPT Fragility with Gemma for comparison. Mini models are unknown sizes and are likely MoE.

not the only piece relevant to handling noise.

Frontier Model Comparison. With both "small" and large GPT models, these results are not truly comparable because we don't have knowledge of additional "reasoning" that the models generate behind the scenes. Some of them are also known to have post-generation classifiers and re-generation, so their algorithms cannot be compared. However, it can help identify possible ceiling scores on Fragility. Figure 9 shows GPT fragility. The largest models are less fragile in the 80's with the "small" GPT models in the 70's, *similar to the best open-source small models*. We include Gemma-12b for comparison. This is significant because the small GPT models are assumed to be larger than these open-source sizes, yet we see similar fragility in them.

Tokenization vs Model Size. To help determine if tokenization (Figure 4) or model (log) size can predict fragility score, we ran a regression analysis. Individually, each shows weak correlation with $R^2 = 0.26$ and 0.42 respectively. However, both predictors in a multiple regression model reveal a strong relationship with fragility: $R^2 = 0.73, F(2, 24) = 33.3, p < .001$.

7 Discussion

What is the core driver of model fragility? Our results reveal that both tokenization and model size are key, but its both together that are most predictive of fragility performance.

Our tokenization studies, illustrated in Figure 5, show that input recognition is important where Gemma's and Qwen's tokenizers are clear winners, performing better on fragility despite fewer parameters. Their tokenizers simply contain more of the lookalikes than the other models. However it's clear that training differences also matter. We see this in Deepseek's Qwen-based tokenizer: while they share tokenization, Deepseek's further training has shifted their embeddings away from ASCII lookalikes, worsening fragility. Future work should investigate why/how.

We also found through statistical tests that the added parameterization within larger models, such as Mistral and OLMo-32b, can allow for reconstruction of a scattered input into a meaningful internal representation. The 32b variant of OLMo, which is also the tokenizer with the *fewest* recognized Unicode characters, has a similar fragility score to the smaller Qwen-4b. Qwen's tokenizer clearly gives it an initial boost to model fragility, but larger parameterization is also capable of catching up later. Our R^2 results confirm this.

These results show that tokenizer recognition of lookalikes is important, and model size can compensate some for missing representation. However, the statistical analysis shows that the two together are most crucial. We recommend intentionally adding Unicode lookalikes to tokenizers as a first step, but urge the need for future research in how to fine-tune for this noise.

We hope this paper leads to further studies in broad fragility, not just for adversarial attacks. Since our Fragility Score is a linear combination of a diversity of tests, it can be customized to different task needs. The Fragility Scorer for LLMs is available with tokenizer analysis code at *github.com/nchambers/fragility*.

Limitations

Unicode character sets are very large and continually expanding with new versions. Our set of hand-identified Unicode lookalike characters is likely not complete, and so these experiments may have missed some characters that would alter the fragility score. However, we intentionally used an external dataset for our experimental set of lookalikes to avoid author bias in what should be included.

This comparison of model fragility does not include all of the closed-source models that are available at the time of writing. Closed-source models are trained on unknown datasets, most privately owned, and architectures are generally not communicated to the scientific community. It is possible that these models contain fundamental advances to assist in character understanding, but determining how or what contributes is not possible by the research community. We include in our results the evaluation of several GPT models, large and small, as data points for the curious reader.

The weighted factors that make up the Fragility Score are chosen to balance character, digit, and overall NLU performance, but these weights are not an empirically motivated decision. Other weights may be more appropriate for any given task, and we believe this flexibility within the Fragility Score is a positive feature.

Ethical Considerations

This paper is a study of how and why LLMs are affected by noisy input at the character level. While much research has focused on the *how*, such as through intentional adversarial attacks, this paper focuses more on the *why* through analysis of model size and tokenization. While identifying these weaknesses carries a risk of dual-use exploitation, a better understanding of these failure modes is essential for building defensive safeguards. We hope that this work is a net benefit to research in assistive systems and more resilient models.

Our study of phone numbers did not intentionally include real numbers, but instead were artificially generated with a random process constrained by the rules of US-based numbers. No human data was used. However, given the saturation of the phone number space, it is possible, even likely, that real numbers were randomly generated and included in the dataset.

Acknowledgments

This work was supported by funding from the Irregular Warfare Technical Support Directorate (IWTSD). The views expressed in this paper are those of the authors and do not reflect the official policy or position of the IWTSD, U.S. Naval Academy, Department of the Navy, the Department of Defense, or the U.S. Government.

References

Nicholas Boucher, Ilia Shumailov, Ross Anderson, and Nicolas Papernot. 2022. Bad characters: Imperceptible nlp attacks. In *Proceedings of the 2022 IEEE Symposium on Security and Privacy (SP)*, pages 1987–2004.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2025. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 23–42.

Cristiano Ciaccio, Marta Sartor, Alessio Miaschi, and Felice Dell'Orletta. 2025. Beyond the spelling miracle: Investigating substring awareness in character-blind language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11361–11372, Vienna, Austria. Association for Computational Linguistics.

Badhan Chandra Das, M Hadi Amini, and Yanzhao Wu. 2025. Security and privacy challenges of large language models: A survey. *ACM Computing Surveys*, 57(6):1–39.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual jailbreak challenges in large language models. In *International Conference on Learning Representations*, volume 2024, pages 24634–24651.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Avia Efrat, Or Honovich, and Omer Levy. 2023. LMentry: A language model benchmark of elementary language tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10476–10501, Toronto, Canada. Association for Computational Linguistics.

Steffen Eger, Gözde Gül Şahin, Andreas Rücklé, Ji-Ung Lee, Claudia Schulz, Mohsen Mesgar, Krishnkant Swarnkar, Edwin Simpson, and Iryna Gurevych. 2019. Text processing like humans do: Visually attacking and shielding NLP systems. 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), pages 1634–1647, Minneapolis, Minnesota. Association for Computational Linguistics.

David A. Haslett. 2025. Tokenization changes meaning in large language models: Evidence from chinese. *Computational Linguistics*, 51(3):785–814.

Linchao He, Dejun Zhang, Long Tian, Han Fei, Mengting Luo, Yilin Chen, and Yiqi Wu. 2018. *Visual-Based Character Embedding via Principal Component Analysis: 4th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2018, Zhengzhou, China, September 21-23, 2018, Proceedings, Part I*, pages 212–224.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *The Ninth International Conference on Learning Representations*, Virtual.

Kyle Hundman, Thamme Gowda, Mayank Kejriwal, and Benedikt Boecking. 2018. Always lurking: Understanding and mitigating bias in online human trafficking detection. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 137–143.

Brynn Knowlton, Jovani Campa, David Solis Gallo, Khalil Dajani, and Nabeel Alzahrani. 2026. Prompt-based jailbreaking of leading LLM chatbots: A survey of attacks and defenses. *IEEE Transactions on Artificial Intelligence*, pages 1–15.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2018. Textbugger: Generating adversarial text against real-world applications. *arXiv preprint arXiv:1812.05271*.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *arXiv preprint arXiv:2004.09984*.

Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. 2024. Drattack: Prompt decomposition and reconstruction makes powerful LLM jailbreakers. *arXiv preprint arXiv:2402.16914*.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning character-level compositionality with visual features. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2059–2068, Vancouver, Canada. Association for Computational Linguistics.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *International Conference on Learning Representations*, volume 2024, pages 56174–56194.

Hongyuan Lu, Haoran Yang, Haoyang Huang, Dongdong Zhang, Wai Lam, and Furu Wei. 2024. Chain-of-dictionary prompting elicits translation in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 958–976, Miami, Florida, USA. Association for Computational Linguistics.

Qibing Ren, Hao Li, Dongrui Liu, Zhanxu Xie, Xiaoya Lu, Yu Qiao, Lei Sha, Junchi Yan, Lizhuang Ma, and Jing Shao. 2025. LLMs know their vulnerabilities: Uncover safety gaps through natural distribution shifts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24763–24785, Vienna, Austria. Association for Computational Linguistics.

Paula Rodríguez Díaz. 2021. Identifying sources of human trafficking within online escort advertisements written in spanish. Master's thesis, Universidad de los Andes.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. In *Proceedings of the Conference on Empirical Methods in Natural Language (EMNLP)*.

Vageesh Saxena, Benjamin Ashpole, Gijs van Dijck, and Gerasimos Spanakis. 2023. IDTraffickers: An authorship attribution dataset to link and connect potential human-trafficking operations on text escort advertisements. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8444–8464, Singapore. Association for Computational Linguistics.

Lucía Summers, Alyssa Shallenberger, John Cruz, and Lawrence Fulton. 2023. A multi-input machine learning approach to classifying sex trafficking from online escort advertisements. *Machine Learning and Knowledge Extraction*, 5:460–472.

Yudian Tian, Dongcheng Li, and Siqing Xue. 2025. A survey of jailbreaking attacks on large language and multimodal models. In *Proceedings of the 2025 8th Artificial Intelligence and Cloud Computing Conference*, pages 142–150.

Omri Uzan and Yuval Pinter. 2026. CharBench: Evaluating the role of tokenization in character-level tasks. *arXiv preprint arXiv:2508.02591*.

Catalina Vajiac, Duen Horng Chau, Andreas Olligschlaeger, Rebecca Mackenzie, Pratheeksha Nair, Meng-Chieh Lee, Yifei Li, Namyong Park, Reihaneh Rabbany, and Christos Faloutsos. 2023. Trafficvis: Visualizing organized activity and spatio-temporal patterns for detecting and labeling human

trafficking. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):53–62.

Yibo Wang, Xiangjue Dong, James Caverlee, and Philip S Yu. 2024. Da^3: A distribution-aware adversarial attack against language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024. Jailbreak attacks and defenses against large language models: A survey. *arXiv preprint arXiv:2407.04295*.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. 2024. Gpt-4 is too smart to be safe: Stealthy chat with LLMs via cipher. *arXiv preprint arXiv:2308.06463*.

Yunxiang Zhang, Liangming Pan, Samson Tan, and Min-Yen Kan. 2021. Interpreting the robustness of neural nlp models to textual perturbations. *arXiv preprint arXiv:2110.07159*.

Brian Zheng, Alisa Liu, Orevaoghene Ahia, Jonathan Hayase, Yejin Choi, and Noah Smith. 2025a. Broken tokens? your language model can secretly handle non-canonical tokenizations. In *Advances in Neural Information Processing Systems*, volume 38, pages 30322–30349. Curran Associates, Inc.

Muyang Zheng, Yuanzhi Yao, Changting Lin, Caihong Kai, Yanxiang Chen, and Zhiquan Liu. 2025b. Mist: Jailbreaking black-box large language models via iterative semantic tuning. *arXiv preprint arXiv:2506.16792*.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

I will give you a unicode character which looks similar to an ASCII letter a-z or A-Z. Your task is to determine what ASCII letter the unicode character most likely resembles. The letter I give is case-sensitive. Question: What ASCII character is most similar to this unicode character: [EXAMPLE A] Answer: A Question: What ASCII character is most similar to this unicode character: [EXAMPLE f] Answer: f Question: What ASCII character is most similar to this unicode character: [CHARACTER] Answer:

Table 2: Prompt used for "What is" letter questions. A similar prompt is used for digits.

You are a phone number decoder. A phone number has been corrupted with unicode characters and punctuation. Your task is to recover just the numerical digits of that phone number. Only respond with the digits of the phone number and no other symbols. Here are a few examples. Q: What are the digits of this phone number: [EXAMPLE 1] A: 2144357735 Q: What are the digits of this phone number: [EXAMPLE 2] A: 7754263233 Q: What are the digits of this phone number: [EXAMPLE 3] A: 5677381287 Q: What are the digits of this phone number [PHONEWITHUNICODE] A:

Table 3: Prompt used for all phone number questions. The examples are mixed with varying digits that have Unicode replacements and/or punctuation noise added, dependent on the specific test.

A Appendix

A.1 Prompts

The prompt templates that we used for the tests in the Fragility Score are presented in Tables 2-6. Several tests have variants that use different numbers of replaced characters or likelihood of noise, but they share the same template prompt. The MMLU baseline test and the MMLU perturbed test, for instance, have the same prompt and just differ in the options text that is presented in the possible answers.

These templates were developed to maximize proper output format and tested across multiple model families to ensure equal opportunity for success.

You are a sentence decoder. Your task is to recover the original sentence from a given string that contains unicode characters that look similar to standard ASCII letters. Only respond with the decoded sentence.

Q: What is this sentence: [EXAMPLE 1]
A: The horse jumped over the fence.
Q: What is this sentence: [EXAMPLE 2]
A: Running is really good for your health!
Q: What is this sentence: [EXAMPLE 3]
A: I enjoy playing video games.
Q: What is this sentence: [QUERY SENT W/ UNICODE]
A:

Table 4: Prompt used for sentence questions.

You are a sentence decoder. Your task is to recover the original sentence from a given string that contains unicode characters that look similar to standard ASCII letters. Follow these three steps: First, break apart the given sentence into individual characters. Second, replace any lookalike unicode characters with their ASCII letters, keeping all others the same. Third, print the sentence in ASCII only.

What is this sentence: [QUERY SENTENCE W/UNICODE]

Table 5: Prompt used for sentence questions with chain-of-thought.

Below is a multiple-choice question. Answer the question with just the letter of the correct answer.

Question: [MMLU QUESTION TEXT]
A. [OPTION A]
B. [OPTION B]
C. [OPTION C]
D. [OPTION D]
Answer:

Table 6: Prompt used for the MMLU tests. The OPTIONs are perturbed or not depending on the test.

A.2 Model Versions and API Usage Cost

The core models evaluated in this work are open-source. We ran the models on two local A100 GPUs, so do not list costs associated with running them. The closed-source GPT models used OpenAI's API interface with the following cost per model to run all tests on the Fragility Score:

Model	Version	API Cost
GPT-4.1	2025-04-14	$8.25
GPT-5.4	2026-03-05	$9.44
GPT-5.5	2026-04-23	$222.25
4.1-mini	2025-04-14	$1.73
4o-mini	2024-07-18	$60.24
5.4-mini	2026-03-17	$2.90
5-mini	2025-08-07	$29.60

These API costs change over time, but are listed as the cost to us at the time of running the experiments.

A.3 Model Scores and Consistency

All scores reported in the main Table 1 are from one run using the Fragility Scorer's static test set. However, to show that the scorer's tests are not tuned to certain performance, we ran Gemma-1b 12 times with different seeds to generate the data. Gemma-1b across the runs averaged an overall fragility score of 36.3 (matching its reported 36) and a standard deviation of 0.7 (95% CI [35.8,36.8], range [35.5,37.8]). There are over 17k tests in the overall fragility score, so this low variance shows that small changes in tests have little effect on the score.

The number of tests in each of the sub-scores of the Fragility Score are given in Table 7. Each row provides the number of tests for a single test set type, but there are multiple variants of some of them to add up to a total of 17k tests.

A.4 Unicode compared to Blanks

The test for 'what is this word' that uses underscores (p_zz_s) is compared to the test with lookalike characters in those same positions. Figure 10 shows the accuracy improvement across all models. Most models show very significant improvement on identifying the word when a lookalike character is given. This shows that models do have the capability of using noisy Unicode, despite its noise hurting other tasks. Performance is worse with Unicode than if it was all plain ASCII, but it's

Type of Test	# of Tests
MMLU	2400
What is this digit?	334
What is this letter?	1099
What is this word?	300
What is this sentence?	299
Phone Number Single	334
Phone Number Multi	200
Phone Number Noise	500

Table 7: The number of tests in each type of sub-score within the Fragility Score.

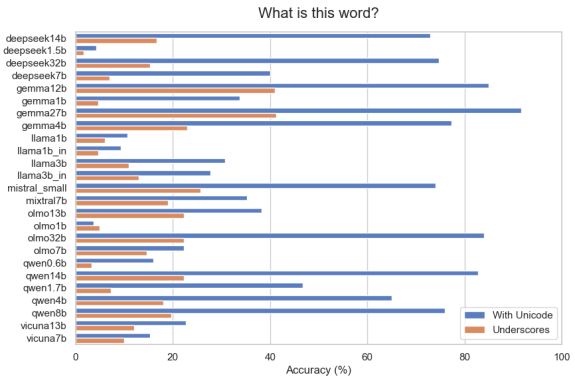


Figure 10: Comparison between the two tests for 'what is this word?'. The first blue bar shows word prediction when Unicode lookalike characters are in the word. The second orange bar uses underscores instead of lookalikes. Most models show significant gain when lookalikes are included.

clear that the lookalike characters provide some information above just treating them as blanks.

A.5 MMLU Experiment Details

We randomly sampled 2400 questions from the 57 subject MMLU benchmark (Hendrycks et al., 2021) for the LLM tests. For the fragility score itself, we measure MMLU accuracy on the 2400 questions without text changes, and then again but with 20% of the characters in the answers swapped with Unicode lookalike substitutes in the answer options. The score is computed as $(1 - acc_{20}/acc)$. If a model struggles to generate the answer in an extractable format, it is marked incorrect, but the ratio assures the score only penalizes on the noise's effect.

For further analysis, we varied the percent of characters beyond just the 20% used in the fragility score. We also randomly replaced by 40, and 60%. These results per model are shown in Figure 11. There is a general trend that the models

Model	Score	Unrec	Unrec-20
OLMo 32b	87	0.2%	2.9%
Gemma 27b	81	1.4%	22.2%
Qwen 4b	80	8.0%	11.5%

Table 8: Unrec=Unrecognized Format. Three of the high-scoring models on the MMLU fragility score are listed here to show their percent of output that was an unrecognized format. These models did not always generate a letter A-D for multiple choice. The Unrec columns show the percent of responses that were unrecognized. Unrec-20 is when 20% of the characters were permuted.

experience a big drop when 20% noise is added, and then continual (but not as large) drops with each 40% and 60% after that.

One potential concern is that the MMLU answers are multiple choice letters A-D, and a model may not immediately generate one of the letters. The prompt that we use is given in Table 6. We structured it in a clear Q/A format that is common to text corpora and used in other MMLU work. We did observe models generating explanations instead of letters, or no answer at all. However, the goal is to measure noise effects, so if a model generates unrecognized formats 20% of the time, that's not of concern to this study. We then measure how noise changes that 20% as well as the correct-to-incorrect flips. A few models and their % blank numbers are given in Table 8. You can see that the small Qwen model actually scores quite high despite an initial high 8% of bad format.

A.6 Punctuation Noise

We include tests for filler noise that does not include Unicode, but instead confounding ASCII punctuation. For phone numbers, this looks like 41..05–5?5-38,53. These tests help compare how model size handles general noise, so that the Unicode lookalike numbers can be better understood in context of a model's overall capability of representing the data. Figure 12 shows each model with different amounts of noise between characters, ranging from 20 to 100%. The 100% value means each character has some filler noise added immediately after it.

The reduction in performance is visible across all models, but the amount of reduction differs. A few of the models are incomplete because the extra tokenization required expands the input context too far to fit on GPUs.

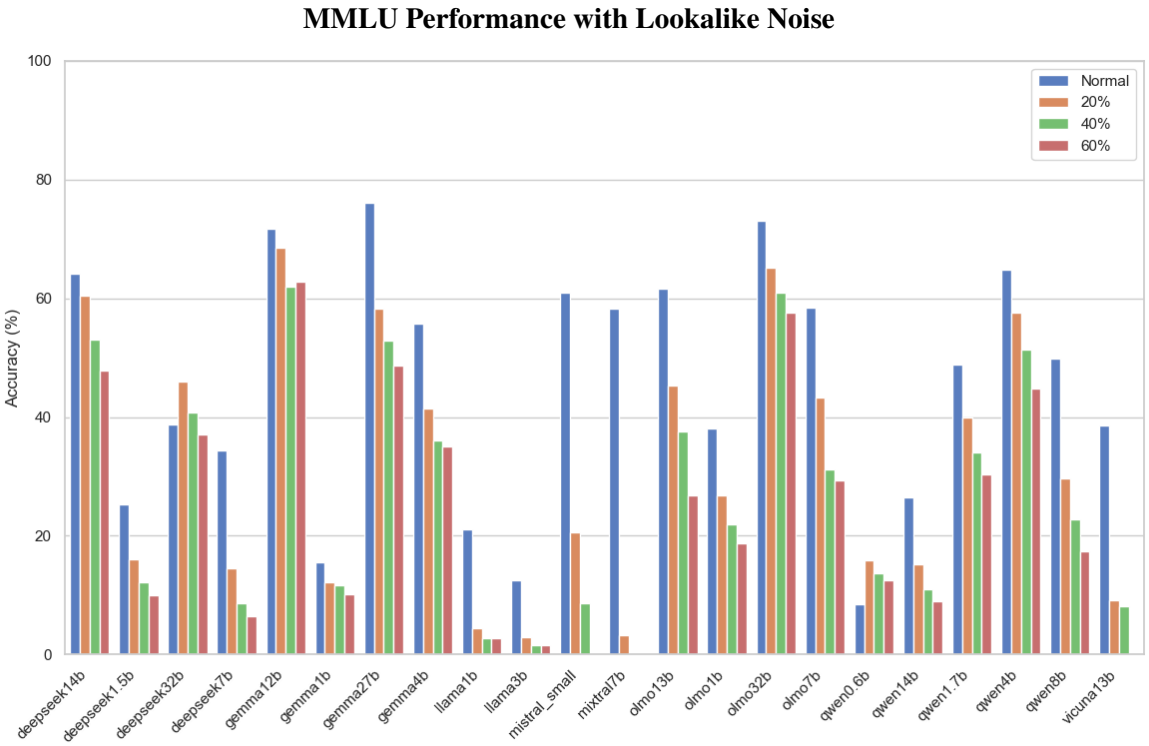


Figure 11: MMLU benchmark results with varying amount of characters replaced by their Unicode lookalikes.

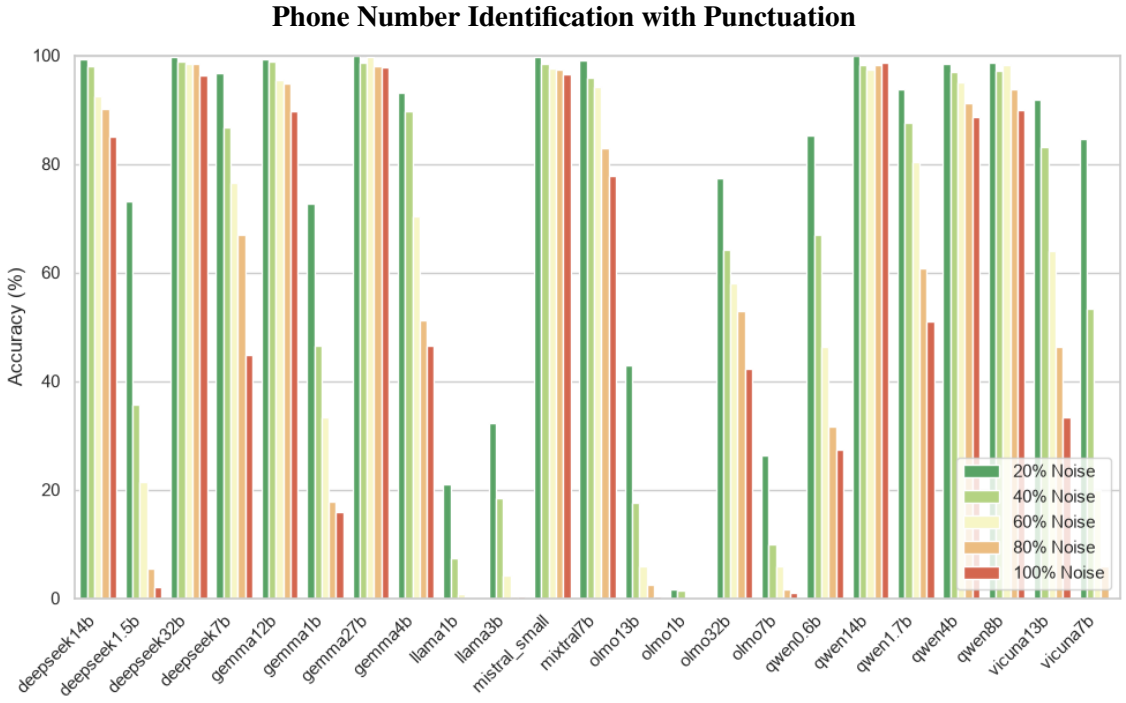


Figure 12: Model results with different amounts of punctuation noise added after each phone number digit.

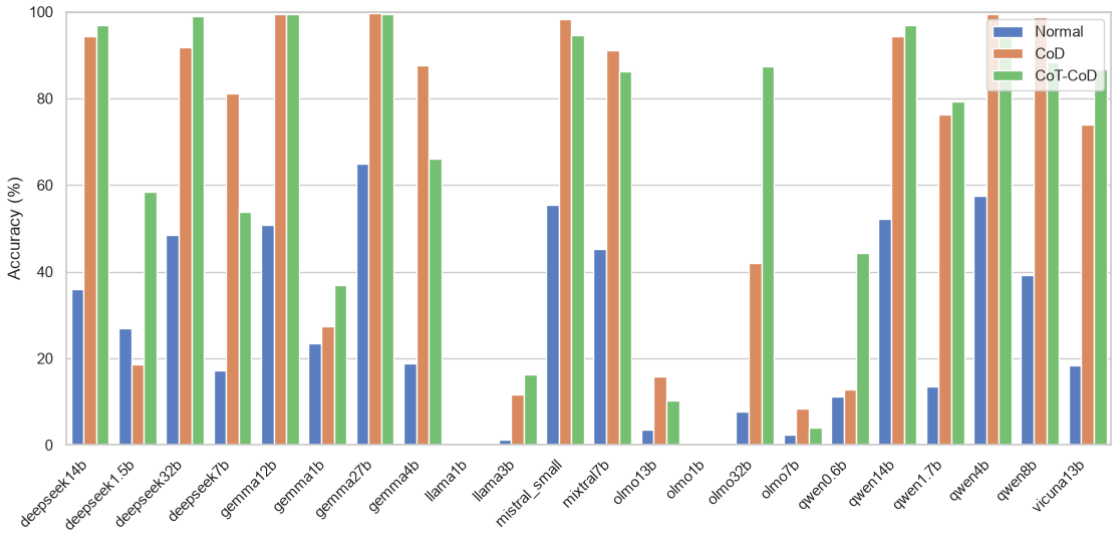


Figure 13: Phone number identification results with chain-of-dictionary (CoD) hints added to the prompts. For each Unicode character in the phone number, its lookalike mapping ('α is a') is added to the prompt. The three bars indicate (1) plain prompt, (2) chain-of-dictionary prompt, and (3) chain-of-dictionary with chain-of-thought prompting. Most models see large jumps with CoD and no additional gains (or even hindrance) from CoT.

A.7 Chain-of-Dictionary Effect

The fragility score includes a subcomponent with a small amount of weight that tests the model's ability to use chain-of-dictionary (CoD) information in the context. If you prompt the model with the gold mapping from lookalike Unicode to its ASCII character, can the model make correct use of the information? We include this in the phone number identification test and Figure 13 shows the per-model improvement with CoD clues added to the prompt. Most models see large jumps with CoD, with the exception of the smallest (1b, 3b, etc.). This suggests that most models have the capacity to understand lookalikes, but they don't have the token representation nor proper training.

Multi-author style detection with stylistic embeddings

Man-Ling Sung, Chak Fai Li, Damianos Karakos, Fred Bernardin, Brian Ulicny
RTX BBN Technologies
Cambridge MA, USA

Abstract

We investigate the use of Large Language Models (LLMs) in multi-author style change detection. Using off-the-shelf LLMs or supervised fine-tuning solely within the target domain fails to yield effective features; for this reason, we propose cross-domain transfer learning with LLM-based embeddings fine-tuned for authorship verification. We fine-tune Mistral-7B embedding model and compare it with DeBERTa, used in the the state-of-the-art (SOTA) systems in PAN-25 multi-author writing style analysis challenge. Results show that DeBERTa and Mistral excel in different user-generated text scenarios. We exploit this diversity by combining both embeddings, leading to more effective style change detection.

1 Introduction

In the domain of user-generated text, authorship detection has long been an important task for verifying the consistency of a user across the text they generate online. Spotting changes in authorship is important for detecting security incidents, such as account hijacking/takeover or plagiarism. In such cases, it is important to detect if multiple users contributed to the writing: the hijacker and the plagiarist may only change *parts* of a document, not the entire document, to evade detection. This problem becomes much harder for online text due to typos, slang, and acronyms. Despite recent advancements in Large Language Models (LLMs), and despite their popularity and success in tackling natural language tasks, they still struggle to directly serve as feature extractors that can effectively capture authorship information. In the PAN-25 multi-author writing style analysis task, DeBERTa (an encoder-based transformer) still performs better as the stylistic embeddings than LLMs – a decoder-based model. However, there are successful attempts at adopting LLMs for cross-domain authorship attribution by extracting stylistic information. One adaptation framework, which we use in our work, is *Learning Universal Authorship Representations* (LUAR) (Rivera-Soto et al., 2021).

Our work explores the use of LLM-based stylistic embeddings for multi-author detection. We want to understand whether **cross-domain authorship representations, learned with a decoder-based embedding model, capture stylistic signals that are complementary to the contextual signals captured by DeBERTa**. We propose a resource-efficient LLM adapted system for effective sentence-level style change detection. We also analyze different types of user-generated text and propose a more versatile system for effective style change detection leveraging both encoder-based and decoder-based models.

2 Problem Definition and Related Work

2.1 Multi-author style change detection

Given a multi-author document, the goal of style change detection is to identify the text positions where the author switches within the document. This task is a form of intrinsic authorship analysis that does not rely on any additional text sources besides the single given document during detection.

We focus on author change detection at the sentence level. For example, in a two-author document $D = \{d_1, d_2, ..., d_N\}$ with N sentences and with a single transition point, the goal is to identify the transition sentence $\hat{i}$ with the property that $\{d_1, .., d_{\hat{i}-1}\}$ is written by author A and $\{d_{\hat{i}}, .., d_N\}$ is written by author B. In general multi-author documents, the goal is to detect all transition sentences where their adjacent sentences are written by different authors.

2.2 Stylistic feature representations

The most common approach to style change detection is the ensemble technique, which embeds a text sentence into representation features for a decision-

Proceedings of the 11th Workshop on Natural User-generated Text, pages 182–194
October 28, 2026 ©2026 Association for Computational Linguistics

making model. Decision-making models can range from simple cosine similarity comparison to machine learning or neural network (NN) classifiers. For representation features, classic methods are based on stylometry (Evert et al., 2017), such as Bag-of-Words (BoW) (Bevendorff et al., 2025), or word graph with carefully selected linguistic features (Boriceanu and Băltoiu, 2025). Current state-of-the-art (SOTA) uses encoder-only transformer models, which are designed to understand contextual meaning of the text, specifically with DeBERTa (He et al., 2021). In PAN-25 multi-author writing analysis task, the two best performing systems both utilize DeBERTa. The first team (Lin et al., 2025b) extracts sentence features with DeBERTa, then trains a NN classifier. Another team (Lin et al., 2025a) uses supervised contrastive learning with joint optimization of classification and contrastive loss, using DeBERTa as the underlying language model. This reflects DeBERTa's strong capability to capture distinguishable stylistic features that a simple classifier can easily learn.

Despite the recent emergence of LLMs, attempts to use pretrained LLMs have not yielded strong results. Llama-3-8B works well at paragraph level (Lv et al., 2024), but struggles at sentence level (Chen et al., 2025). In sentence level detection, whether it is trying to extract sentence representation with pre-trained LLM, e.g. Claude (Schmidt et al., 2025), or fine-tuned Llama-3, both struggle to see competitive improvement beyond DeBERTa. One limitation on fine-tuning LLM is the vast amount of data required to effectively adapt the model to the target domain. As far as we know, there is no prior work in using existing LLM embeddings for such a task.

2.3 LLM adaptation for stylistic embeddings

With limited in-domain data for model adaptation, the focus has been on cross-domain transfer. In authorship verification, a universal framework – Learning Universal Authorship Representations (LUAR) – is proposed for effective domain transfer (Rivera-Soto et al., 2021). It adopts a contrastive-learning framework to produce style-focused embeddings on top of a language model text embeddings, with max-over-time pooling, which is robust across topics and domains. The stylistic features trained on human authors, can also be transferred to detect machine-generated text from human (Rivera Soto et al., 2024; Kandula et al., 2025).

Efforts have also been made to enable an LLM to perform specific tasks where it normally fails. This approach typically relies on carefully crafted, task-specific data to fine-tune the model. For instance, cross-domain authorship attribution with LLMs can be framed as a retrieval task (Agarwal et al., 2025). Along these lines, Linq-embed-mistral (Choi et al., 2024) is a fine-tuned version of a Mistral model specifically for text retrieval.

3 Methodology

Our system performs two steps, a LLM is used for feature extraction, followed by sentence-level classification with a neural network classifier. The system diagram is presented in Figure 1.

3.1 Feature Extraction Model

We adopted a training methodology in Kandula et al., 2025, a LUAR framework that fine-tunes the Linq-embed-mistral model with contrastive learning. Mistral-7B is fine-tuned with LoRA and low-bit quantization. To extend its capability, the model is further fine-tuned on open-source datasets to comprehensively cover multiple aspects of user-generated text. Datasets span diverse domains, including social media discussions, reviews and comments, email and narratives (see Appendix A for details). We also explored replacing BERT to other white-box models in LUAR, such as Qwen2 (An et al., 2024) and multilingual-e5-large (mE5) (Wang et al., 2024). Linq-embed-mistral (LE-Mistral-7B) performs the best on all PAN-25 tasks, as can be seen in Table 1.

PAN-25 (Valid)	Easy	Medium	Hard
Qwen2-1.5B	0.872	0.798	0.703
LE-Mistral-7B	**0.904**	**0.834**	**0.734**
mE5	0.862	0.772	0.685

Table 1: Results on PAN-25 validation set adopted with LUAR framework. Sentence-level features is used. For an explanation of the easy/medium/hard conditions, please read Section 4.1.

3.2 Neural Network Classifier

We train a binary classifier on the sentence embeddings along with their absolute difference. The architecture consists of fully connected multi-layer perceptron (MLP) layers, with ReLU activation and a dropout layer of rate 0.1. It is trained with batch size 64, learning rate 1×10^{-5}, AdamW optimizer and BCEWithLogitsLoss.

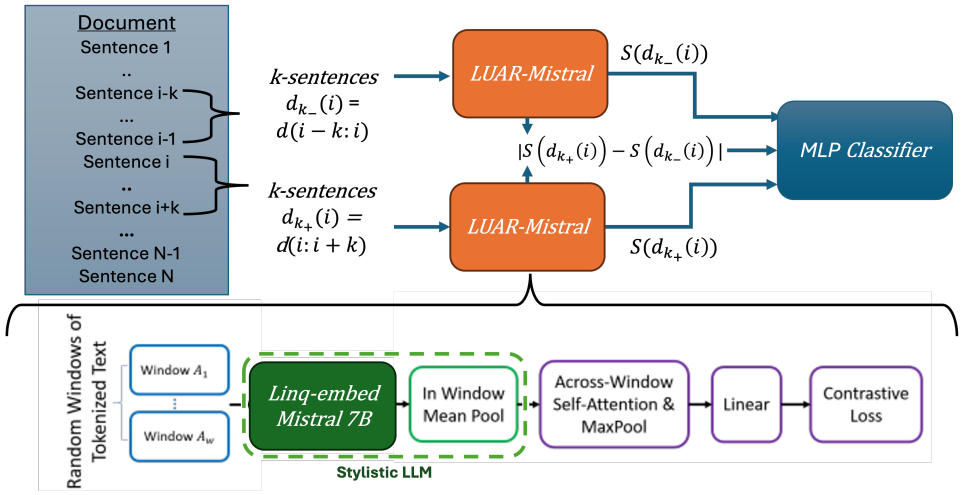


Figure 1: The proposed system. LUAR-Mistral generates embeddings from k sentence-pairs, where k is the number of sentences. This figure illustrated the single sentence case where k=1. The MLP classifier takes in both sentence embeddings and their absolute difference to decide if authorship change happens in between. The sentence embeddings come from a modified LUAR framework which replaces BERT with Linq-embeded Mistral. The framework is fine-tuned to optimize the contrastive loss of same and different authors.

The classifier, in its baseline form, uses pairs of adjacent sentence features as input, as done in the systems submitted in PAN-25. However, we believe that incorporating additional context facilitates detection, especially in short sentences with limited stylistic information (commonly found on social media). We therefore explore a second feature variation, which includes additional features spanning $2k$ sentences, separated in the middle by the hypothesized transition sentence. In the first case (just pairs of adjacent sentences) we use a 2-layer MLP, presented in equation (1); in the second case (k + k sentences) we use a 3-layer MLP, presented in equation (2). The second approach integrates structural information, enabling the model to look beyond single sentences. The decision of the classier $\hat{y}$, is computed as follows for each MLP classifier.

The 2-layer MLP that takes in a pair of sentence embeddings and their absolute difference:

$$\hat{y} = \sigma(\mathbf{W_2} \cdot Dropout(ReLU(\mathbf{W_1} \cdot \mathbf{S}_{LM})) \quad (1)$$

where $\mathbf{W_2} \in \mathbb{R}^{1\times 0.5f}$, $\mathbf{W_1} \in \mathbb{R}^{0.5f\times 3f}$ are learnable parameters. $\mathbf{S}_{LM}$ denotes the input embedding from the fine-tuned language model and f is the embedding size. In the case of Linq-embed Mistral 7B, $f = 1536$, and in the case of Qwen2 and multilingual-e5-large, $f = 1024$.

The 3-layer MLP, which on top of the features fed into the 2-layer MLP, takes in an additional set of multi-sentence embedding pairs and their difference:

$$\hat{y} = \sigma(\mathbf{W'_3} \cdot Dropout(ReLU(\mathbf{W'_2} \cdot Dropout(ReLU(\mathbf{W'_1} \cdot \mathbf{S}_{LM})))) \quad (2)$$

where $\mathbf{W'_3} \in \mathbb{R}^{1\times 0.5f}$, $\mathbf{W'_2} \in \mathbb{R}^{0.5f\times 3f}$, $\mathbf{W'_1} \in \mathbb{R}^{3f\times 6f}$ are learnable parameters. $\mathbf{S}_{LM}$ and f are described above.

4 Experiments and Analysis

4.1 Dataset

We use the PAN-25/26 multi-author writing style analysis datasets. This problem focuses on detecting transition sentences where the change of authorship happens. PAN-25 (Bevendorff et al., 2025) combines Reddit comments into documents. The dataset includes discussions about world news, politics, history, and legal advice. PAN-26 (Bevendorff et al., 2026) contains user-written stories that reuse fictional characters and settings from existing works. Both datasets are split into 3 tasks of various difficulty levels – easy, medium and hard. For the easy task, each document contains multiple topics written by different authors, allowing detection to be framed around identifying topic changes. For the medium task, documents are topically coherent, while stylistic similarity between sentences is moderate. For the hard task, the documents contain a single topic from various authors who are stylistically similar. Our primary interest lies in the medium and hard tasks, with a focus on detecting

stylistic differences within the same topic. Notably, PAN-26 is much larger than PAN-25, with 10 times more sentences and 2-3 times more documents. Detailed analysis is in Table 4.

4.2 Experiment Design

The classifiers are trained on PAN training sets for 20 epochs and scored against the PAN validation sets with decision threshold tuning to maximize the F1-score. The model checkpoint that gives the highest F1 on validation set is used for evaluation. We evaluated three versions of our system: 1) the baseline Linq-Embed-Mistral-7B, 2) **LUAR-Mistral**: Linq-Embed-Mistral-7B fine-tuned with public data (shown in Appendix A) using LUAR, 3) **LUAR-Mistral-PAN**: LUAR-Mistral further fine-tuned with provided PAN training set. We use DeBERTa-v3a to cluster the sentences from the target training set, and selected 40k sentences uniformly from the clusters to create sentences pairs for LUAR. We extract the features at sentence-level and multi-sentence-level, subsequently fitting them to their respective classifiers. For multi-sentence-level setup we sweep the number of sentences $k = [2...7]$.

We reproduce the challenge baseline (BoW with cosine similarity), as well as DeBERTa-Classifier and SCL-DeBERTa. We did not reproduce the LLM-based systems from PAN-25 due to their overlapping nature with the Mistral model evaluated in this work. Both the DeBERTa and Mistral systems were executed on a single NVIDIA A6000 GPU.

4.3 Results and Analysis on PAN-25/26

We follow the PAN challenge that measures the performance in Macro-F1 score, the results are presented in Table 2. DeBERTa achieves the highest performance on PAN-25 easy/hard tasks and PAN-26 easy task, while the Mistral systems excel in PAN-25 medium and PAN-26 medium/hard tasks. Extending neighboring sentences yields better performance, especially for PAN-25 hard and PAN-26 medium tasks. The gain from further fine-tuning LUAR-Mistral with carefully crafted domain data is minimal compared to increasing the context window for the sentence embeddings.

Surprisingly, even though the PAN-26 training set is much larger than PAN-25, detection accuracy drops significantly on medium/hard tasks. The systems struggle to detect authorship change even with more training data. We suspect the system can learn to detect transition words instead of the semantic or stylistic difference within sentences in PAN-25 to yield such high accuracy (see examples in Appendix B). It is known that transformer based models, such as DeBERTa perform better than LLMs in disentangling detection (Azuma et al., 2025).

5 Comparing DeBERTa and Mistral

5.1 Experiment on HIATUS Research Set

To validate our hypothesis, we run the same multi-author detection task on the HIATUS Research Datasets (HRS) (IARPA, 2024). It consists of five genres - board-game reviews, instructions, journalism, posts from literature and STEM forums. We construct our own multi-author documents by randomly selecting 10 to 50 sentences from source documents within the same genre. Specifically, we generate documents containing writings from two to four authors, with each author appearing exactly once within each document. We train 2-layer MLP classifiers using DeBERTa and LUAR-Mistral embeddings respectively. For the extended context variant of Mistral, we set the window size to $k = 4$ and employ a 3-layer MLP classifier. We used an 80/20 train/test split within the same genre.

Table 3 shows that DeBERTa and LUAR-Mistral perform better on different document genres. DeBERTa significantly outperforms on the STEM forum, where the discussion is conversational and features domain-specific terminology and slang, similar to PAN-25. Mistral significantly outperforms on instructional documents, where the writing format is highly standardized, with clear, step-by-step instructions, and pre-defined technical terms and verbs are used. This makes it harder for style change detection. This is also the hardest HRS set that has the lowest F1 for both models, however Mistral's degradation is less than DEBERTa's. Otherwise, DeBERTa slightly outperforms on journalism, which focuses on reporting specific events, while Mistral performs better on boardgames reviews and literature forum, which tend to be more subjective and expressive.

5.2 System Combination

We concatenate DeBERTa and LUAR-Mistral embeddings to train the classifier. The reason for such a combination is to leverage the characteristics of bi-encoder transformer-based features and those of forward LLM-based features to produce a more generalized multi-author transition detection capability. It is a simple system combination

Task	PAN 2025 (test)			PAN 2026 (valid)		
Model	**Easy**	**Medium**	**Hard**	**Easy**	**Medium**	**Hard**
Baseline (Bevendorff et al., 2025)	0.439	0.44	0.453	0.410	0.489	0.442
Llama-3 (4-bit) ft (Chen et al., 2025)	0.461	0.583	0.484	/	/	/
Claude-3.7-sonnet (Schmidt et al., 2025)	0.856	0.818	0.661	/	/	/
SCL-DeBERTa (Lin et al., 2025a)	0.950	0.812	0.827	0.980	0.550	0.489
DeBERTa-Classifier (Lin et al., 2025b)	**0.958**	0.822	0.829	**1.000**	0.633	0.735
Linq-Embed-Mistral-7B (Choi et al., 2024)	0.928	0.790	0.731	0.997	0.613	0.676
LUAR-Mistral	0.938	0.830	0.762	0.992	0.682	0.740
+ k sentences	0.946	0.845	0.772	0.992	0.707	0.743
LUAR-Mistral-PAN	0.936	0.830	0.763	0.990	0.650	0.736
+ k sentences	0.947	0.844	0.766	0.992	0.702	0.741
System Combination (Deberta+LUAR-Mistral)						
Deberta + LUAR-Mistral	0.952	0.846	0.828	**1.000**	0.689	0.764
+ k sentences	**0.958**	**0.858**	**0.844**	1.000	**0.718**	**0.769**

Table 2: Results on PAN-25/26. The top table contains results with previous systems, second table is with our own systems, bottom table is with combined systems. The results for LLama-3 and Claude-3.7-sonnet are copied from (Zangerle et al., 2025), while the rest are our own reproduction. The best overall result in each column is bolded, while the best single system result is underlined.

Genre	DeBERTa	Mistral	+k-sent
B.G. Reviews	0.710	0.731	**0.734**
Instruction	0.609	0.686	**0.694**
journalism	**0.809**	0.780	0.798
LIT. forum	0.776	**0.790**	0.779
STEM forum	**0.913**	0.744	0.755

Table 3: F1 scores on the five HRS genres. Notations: B.G.- Boardgame, LIT. - Literature.

method and does not require large-scale modifications to the underlying feature extractors. We trained the following configurations: (1) 2-MLP layer classifier trained on DeBERTa features concatenated with sentence-level LUAR-Mistral features, (2) 3-MLP layer classifier trained on DeBERTa features concatenated with multi-sentence-level LUAR-Mistral features. Results are presented in the lower section of Table 2. The combined system achieves competitive performance against standalone benchmarks, even outperforming them on the PAN-25 and PAN-26 medium and hard tasks.

6 Conclusion

We explore cross-domain adaptation of an LLM using the LUAR framework for authorship attribution. Experiments show that such LLM embeddings can capture stylistic information for multi-author detection, and using multiple-sentence features yields better detection than single-sentence features. We also experimented with multiple embedding models across various types of user-generated text, and proposed a more all-rounded system by combining DeBERTa and LLM embeddings for effective style change detection.

7 Limitations

A key limitation of pre-trained LLMs is the lack of control over their internal data, making it difficult to verify if our experimental datasets were included in their training. Additionally, the response time of the combined system remains slow due to the integration of multiple features and use of LLM, even though LoRA improves the robustness of the Mistral model.

Our initial experiments reveal that the decoder-based model and the encoder-based model excel in different genres, which suggests the possibility that these models are capturing different linguistic dimensions. More in-depth study is needed to validate this hypothesis and determine the reasons behind these variations. A more detailed study on style-change datasets, especially machine generated text, is also needed to better understand the domain-dependent requirements for effectively evaluating stylistic features. Current multi-author detection datasets fail to reflect collaborative scenarios where authors intend to produce cohesive writing with smooth transitions. Further work is required to bridge the gap with real-world use cases.

8 Acknowledgment

The authors would like to thank the BBN ALERT team and the NLP reading group for the useful discussions. This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2022-22072200003. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Shantanu Agarwal, Joel Barry, Steven Fincke, and Scott Miller. 2025. Cross-genre authorship attribution via llm-based retrieve-and-rerank. *arXiv preprint arXiv:2510.16819*.

Yang An, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Daichi Azuma, René Meléndez, Michal Ptaszynski, Fumito Masui, Lara Aslan, and Juuso Eronen. 2025. Svm, bert, or llm? a comparative study on multilingual instructed deception detection. *AI*, 6(9):239.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. In *Proceedings of the international AAAI conference on web and social media*, volume 14, pages 830–839.

Janek Bevendorff, Daryna Dementieva, Maik Fröbe, Bela Gipp, André Greiner-Petter, Jussi Karlgren, Maximilian Mayerl, Preslav Nakov, Alexander Panchenko, Martin Potthast, and 1 others. 2025. Overview of pan 2025: Generative ai detection, multilingual text detoxification, multi-author writing style analysis, and generative plagiarism detection. In *European Conference on Information Retrieval*, pages 434–441. Springer.

Janek Bevendorff, Maik Fröbe, André Greiner-Petter, Andreas Jakoby, Maximilian Mayerl, Preslav Nakov, Henry Plutz, Martin Potthast, Benno Stein, Minh Ngoc Ta, and 1 others. 2026. Overview of pan 2026: Voight-kampff generative ai detection, text watermarking, multi-author writing style analysis, generative plagiarism detection, and reasoning trajectory detection. In *European Conference on Information Retrieval*, pages 225–232. Springer.

I Boriceanu and A Băltoiu. 2025. Style change detection using graph and structural-linguistic features for multi-author writing analysis. *Working Notes of CLEF*.

Dongjie Chen, Jijie Li, and Haoliang Qi. 2025. Llama-3 with 4-bit quantization and ia 3 tuning for multi-author writing style analysis. *Working Notes of CLEF*.

Chanyeol Choi, Junseong Kim, Seolhwa Lee, Jihoon Kwon, Sangmo Gu, Yejin Kim, Minkyung Cho, and Jy-yong Sohn. 2024. Linq-embed-mistral technical report. *arXiv preprint arXiv:2412.03223*.

Stefan Evert, Thomas Proisl, Fotis Jannidis, Isabella Reger, Steffen Pielström, Christof Schöch, and Thorsten Vitt. 2017. Understanding and explaining delta measures for authorship attribution. *Digital Scholarship in the Humanities*, 32(suppl_2):ii4–ii16.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

IARPA. 2024. IARPA HIATUS Phase 1 HRS Data. Available to HIATUS performer teams via the ARLIS HIATUS Portal. Intelligence Advanced Research Projects Activity (IARPA). Released by the Data Access Center (DAC).

Hemanth Kandula, Chak Fai Li, Haoling Qiu, Damianos Karakos, Hiu Mn, Thien Huu Nguyen, and Brian Ulicny. 2025. Bbn-u. oregon's alert system at genai content detection task 3: Robust authorship style representations for cross-domain machine-generated text detection. In *Proceedings of the 1st Workshop on GenAI Content Detection (GenAIDetect)*, pages 358–364.

Aashita Kesarwani. 2019. New york times comments. Kaggle.

Mike Kestemont, Enrique Manjavacas, Ilia Markov, Janek Bevendorff, Matti Wiegmann, Efstathios Stamatatos, Martin Potthast, and Benno Stein. 2020. Overview of the cross-domain authorship verification task at pan 2020. In *CLEF (Working Notes)*.

Aleem Khan, Elizabeth Fleming, Noah Schofield, Marcus Bishop, and Nicholas Andrews. 2021. A deep metric learning approach to account linking. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5275–5287.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer.

Kaichuan Lin, Chang Liu, Fuchuan Ye, and Zhongyuan Han. 2025a. Scl-deberta: multi-author writing style change detection enhanced by supervised contrastive learning. *Working Notes of CLEF*.

Xiaocan Lin, Chang Liu, Xianbing Duan, and Zhongyuan Han. 2025b. Team wqd at style change detection in multi-author writing: A deep learning approach based on deberta. *Working Notes of CLEF*.

Jiajun Lv, Yusheng Yi, and Haoliang Qi. 2024. Team fosu-stu at pan: Supervised fine-tuning of large language models for multi author writing style analysis. In *CLEF (Working Notes)*, pages 2781–2786.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)*, pages 188–197.

Francisco Rangel and Paolo Rosso. 2019. Overview of the 7th author profiling task at pan 2019: bots and gender profiling in twitter. *Working notes papers of the CLEF 2019 evaluation labs*, 2380:1–7.

Francisco Rangel, Paolo Rosso, Martin Potthast, and Benno Stein. 2017. Overview of the 5th author profiling task at pan 2017: Gender and language variety identification in twitter. *Working notes papers of the CLEF*, 48.

Rafael Rivera Soto, Kailin Koch, Aleem Khan, Barry Chen, Marcus Bishop, and Nicholas Andrews. 2024. Few-shot detection of machine-generated text using style representations. In *International Conference on Learning Representations*, volume 2024, pages 29392–29412.

Rafael A Rivera-Soto, Olivia Elizabeth Miano, Juanita Ordonez, Barry Y Chen, Aleem Khan, Marcus Bishop, and Nicholas Andrews. 2021. Learning universal authorship representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 913–919.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205. Stanford, CA, USA.

Gleb Schmidt, Johannes Römisch, Mariia Halchynska, Svetlana Gorovaia, and Ivan P Yamshchikov. 2025. Team" better_call_claude": Style change detection using a sequential sentence pair classifier. *arXiv preprint arXiv:2508.00675*.

Yanir Seroussi, Ingrid Zukerman, and Fabian Bohnert. 2014. Authorship attribution with topic models. *Computational Linguistics*, 40(2):269–310.

Efstathios Stamatatos, Francisco Rangel, Michael Tschuggnall, Benno Stein, Mike Kestemont, Paolo Rosso, and Martin Potthast. 2018. Overview of pan 2018: Author identification, author profiling, and author obfuscation. In *International conference of the cross-language evaluation forum for european languages*, pages 267–285. Springer.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Yelp. 2021. Yelp dataset. Kaggle. Accessed: 2026-07-13.

Eva Zangerle, Maximilian Mayerl, Martin Potthast, and Benno Stein. 2025. Overview of the multi-author writing style analysis task at pan 2025. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2025), Madrid, Spain, 9-12 September, 2025*, volume 4038 of *CEUR Workshop Proceedings*, pages 3568–3574. CEUR-WS.org.

A System design and decision

A.1 List of datasets to train LUAR-Mistral

We trained our LUAR-Mistral with opensource datasets listed below:

- Social Media Platform:
  - Reddit (Khan et al., 2021; Baumgartner et al., 2020)
  - Twitter (Rangel et al., 2017; Stamatatos et al., 2018; Rangel and Rosso, 2019)
- Reviews and Comments:
  - IMDB (Seroussi et al., 2014)
  - Yelp (Yelp, 2021), Amazon (Ni et al., 2019)
  - New York Times (Kesarwani, 2019)
- Formal Communication:
  - Email (Klimt and Yang, 2004)
- Narratives and Stories:
  - Blog (Schler et al., 2006)
  - Fan-Fiction (Kestemont et al., 2020)

B Analysis of PAN-25 and PAN-26 Data

Table 4 presents a overview of the PAN-25/26 datasets, on the number of documents, sentences and average word per sentence. This gives an idea of how much data the system is exposed in training, and how different tasks deviate from each other. Notice the PAN-26 medium task is much larger than the other tasks, but surprisingly with lower F1 overall.

We also list a sample for each of the task and provide a brief analysis on the document characteristics in Tables 5, 6, 7, 8, 9, 10. The transition sentences are highlighted in blue.

PAN-25				
Task	Set	# Doc	#Sent	#Word/Sent
Easy	Train	8.4k	48k	18.31
	Valid	1.8k	10k	18.42
	Test	1.8k	10k	18.55
Medium	Train	8.4k	59k	18.91
	Valid	1.8k	13k	19.00
	Test	1.8k	13k	18.90
Hard	Train	8.4k	51k	20.07
	Valid	1.8k	11k	20.49
	Test	1.8k	11k	19.87
PAN-26				
Task	Set	# Doc	#Sent	#Word/Sent
Easy	Train	21k	549k	29.90
	Valid	4.5k	117k	29.84
Medium	Train	21k	1299k	27.03
	Valid	4.5k	279k	27.01
Hard	Train	21k	590k	28.01
	Valid	4.5k	127k	27.87

Table 4: Detail of the train, valid and test sets for the 3 tasks in PAN25 and PAN26, presented in number of documents, sentences and average number of word in each sentence.

PAN-25 Easy (4 authors)
In general, be courteous to others.
Debate/discuss/argue the merits of ideas, don't attack people.
Personal insults, shill or troll accusations, hate speech, any suggestion or support of harm, violence, or death, and other rule violations can result in a permanent ban.
Correcto.
They refer to them more easily as animals or a disease.
Only suitable to be wiped out and systematically destroyed.
Like stray dogs with rabies.
r/politics is currently accepting new moderator applications.
If you want to help make this community a better place, consider !
Exactly.
The response should have been to impound every bus, arrest, charge, and hold every driver on human trafficking and kidnapping charges and issue a warrant for the arrest of Abbott and any staff of his or the bus lines identified as involved.
Yup.
Wilhoit's law: "Conservatism consists of exactly one proposition, to wit: There must be in-groups whom the law protects but does not bind, alongside out-groups whom the law binds but does not protect.".

Table 5: Problem 10 of PAN-25 easy training set. Beside the stylistic changes, transition can be identified easily with words that indicate the response to the previous users – "Correct", "Exactly" and topic header – "r/politics".

PAN-25 Medium (2 authors)
Other players will get called up and the world cup is the biggest stage to present yourself.
Players get huge contracts after world cups because they played well.
It simply won't happen.
Major FAs like Brazil, Germany, France, England and so on could have joined forces and declined to play, that would have been an option.
But those are most of the time on the same level of corruption.
I don't blame the players, it would have been a Kaepernick situation just in a different sport.
Their name would have been dragged through the mud and FIFA would have done anything in their power to discredit them.
Football is dirty behind the scenes and I don't think it's as easy to drop out of the worldcup as a player as you think without facing major repercussions.
And by "terrible situation", he specifically means 6,500 deaths so far and counting.
For a ** soccer match.
FIFA has blood on its hands.
All those workers were real people with real lives and families.
They were poor and brown, but they were regular ass human beings at the end of the day.
How the fuck could they let this happen?
Would they have ignored even 2 or 3 deaths in America or Canada?
Forget 6,500.

Table 6: Problem 10 of PAN-25 medium training set. The transition sentence is not trivial to detect, but change of style is observed in neighboring sentences with the use of "soccer" and "football" (highlighted in orange manually) that describe the FIFA event.

PAN-25 Hard (4 authors)
Soooo. . . according to the article, the Chinese army started scheduled training drills in the area this morning, and all of a sudden, fishermen saw a ufo?
The US just confirmed the others were balloons too.
Schumer said it on an interview on ABC this morning.
They haven't blamed them on China though.
Not officially.
But he spun the announcements of the balloons into talking about China's balloon spy program.
there are only so many times you can use the word balloon in a pentagon briefing before people start to laugh it off.
Ah the old "look we have spy balloons invading us too, so those other spy balloons couldn't have been ours."
trick.
Of course they are forgetting that they already demanded their downed spy balloons returned to them, the other day.
Yeah they weren't the same as the PRC balloon, however they were still balloons.
I don't see why the general would call them balloons if they weren't.
Already have read your article.
It states these objects are not similar to the original PRC balloon, however they are still balloons according to Schumer and the Canadian General.

Table 7: Problem 10 of PAN-25 hard training set. The topical difference is not obvious, however transition words such as "Ah", "Yeah" that indicate the response to the previous users is still presented in the transition sentence.

PAN-26 Easy (3 authors, 5 transitions)
“Y/N Look out” You quickly turned your head only to see one of the Devaronians aim their blaster towards you. Instead of answering, you glared daggers at him, which caused him to snicker as he made his way towards you. A quick glance at your fuel reserves told you that would have ended in disaster, so, with a fake smile plastered on your face and a bowed head for anyone in uniform, you quickly explained that you were on a tour of First Order allied planets as a gift from your parents for your eighteenth birthday, and that you had been cruising around before realizing you would need to land as soon as possible to refuel. “Y/N, I... I haven’t been completely honest with you, and I need to tell you this before...” “W-what do you mean, Poe? “Poe—“ “Please, Y/N, let me finish or I will never get this out,” You stayed silent and Poe continued. “You’re going to be a fun one to break.” As he finally released you, your gaze went straight down to Poe who continued to lay motionless on the ground. And in that moment, as you looked up into Poe’s dark eyes, you knew that everything was going to be okay. Y/N, are you okay?” You realized Poe was behind you and you tried to look back towards him, only to be able to see the side of his face through your peripherals. “You really are somethin’, you know that Dameron?” Poe smiled at you in a way that you somehow knew was only reserved for you as he smoothed your Y/H/C out of your face, leaning forward to place a feather light kiss on your forehead. “Just someone who is crazy about you, sweetheart.” You sighed happily and leaned into Poe’s touch—wrapping your arms to lessen the little distance that was between the two of you. “Y/N... hang on sweetheart, I’m coming for you.” The dust cleared and you could see that it was Poe getting closer to you, his hand reaching out. Didn’t I tell you I would come up with a plan to get us out of here?” You let out a breathy laugh and looked up to see Poe smiling slightly. This is getting little inten—“ “Already ahead of you.” Poe waved the com-link in your face and you smirked. The resulting explosions had you thinking that these were some sort of weapons manufacturers, and suddenly the knot in your stomach made sense; whatever this planet was used for obviously dealt in death. Tanned skin, dark curly hair, and warm brown eyes smile down at you while he holds out his hand for you to shake. You tried to rush into your ship, hoping to get on the comms and call out to the X-Wings that you were a friendly, but your ship was hit with a blast that sent you flying; unlike the Troopers that had been helping you, you got tossed to an open field – they collided with a nearby wall with a sickening crunch.

Table 8: PAN-26 Easy train set problem 10. Due to the nature of consistency in stroy telling, the change of topic is different than PAN-25 without transition words.

PAN-26 Medium (20/118 lines)
Steve goes to Pepper for help because even though he's only just met her and he still doesn't know her very well, he can tell that she's a good person with a kind heart and that she won't say no to any reasonable request.
He also knows that she is the sort of person who can get things done and that she will do the things that need to be done with a minimum of fuss and bother.
They meet for coffee in Bryant Park on a sunny day in March and Pepper listens quietly as Steve outlines what he wants and why.
Pepper doesn't blink an eye, just whips her StarkPad out of her purse and starts tapping away as they brainstorm ideas.
Bucky is the most alive person Steve has ever known and the idea of putting his name on a grave is unimaginably wrong.
He immediately vetoes her suggestion of a memorial stone in Arlinon because he just cannot bear the thought of a tomb of any type.
In the end Steve decides that he doesn't want to plant a sapling and chooses one of the heritage trees; a big white oak that curves majestically toward the sun and that has a canopy as big as a small swimming pool.
When they hit upon the commemorative tree program in Prospect Park they both know that they've found the answer.
But a tree – a living thing that could grow and flourish for more than a life time - a tree is just perfect.
Steve does not deliberately seek out the tree, but he does sometimes glimpse it out of the corner of his eye if his daily run takes him that way.
But after he presses them for more information they estimate that it might have been planted sometime in the early twentieth century and that settles it in Steve's mind.
The park authorities aren't entirely sure how old it is because the records are inconclusive and unclear.
There are no signs anyone else has been down this way recently, and when they get to the first village, they are told that no one else has come down from the mountains in weeks.
There are more bad days than good - tears intermingled with stony silence or fits of rage so vast they make a tsunami look tame.
And every day he wakes up and hopes that this one will be better than the last, that he'll open his eyes to find that everything has shifted for the better.
Despite his not wanting to make a fuss and his desire to keep things as simple as possible, he's actually insanely pleased to see them troop through the door and settle at the table.
In late October they hold a simple ceremony to unveil the plaque, just Steve, Pepper and Tony in attendance.
Afterwards they go to a local steak house for prime rib and beer and somewhere around 3pm the other Avengers start to trickle in.
They end up taking over a whole corner of the restaurant, ordering more food and more beer and being generally themselves, which is actually kind of scary when Steve thinks about it closely enough.
But the owners can't really complain once Tony has slapped his credit card on the bar, ordering them to charge everything to him, and they stand back and let them carry on regardless.
...

Table 9: Sub-document of PAN-26 Medium train set problem 10. Due to the nature of consistency in stroy telling, the change of topic is less obvious than PAN-25. But with careful reading we can notice the location changes from park (city) to village and mountain (remote) area at the adjacent transition sentences.

PAN-26 Hard (2 authors, 1 transition)
Your tears flowed freely after that and you leaned forward to press your lips to his forehead, clinging lightly to him but avoiding the injuries from the battle.
You sniffed, not even realizing until then that you had begun to cry, as he tried to comfort you, tears of grief and guilt clinging to his lower lashes.
“You nearly died Thorin,” you whispered, turning your palms to cup his bearded jaw, “You nearly died and I almost didn’t get to tell you. . . ”.
“Well now Thorin, what’s going on because I can tell that it’s not anything to do with an illness.” Oin was very matter of fact when it came to these things.
“How did you know?” Thorin wasn’t startled by his friend and general, the dwarf did live in the same wing as the king, in fact most of the company did.
Thorin looked over at his general, his eyebrow raised in question, the gruff dwarf turned red and nodded.
She looked at Dwalin who nodded and took over for her, she nodded back and moved to follow Thorin as he walked out of the training area.
Dwalin turned to the woman behind him “Good on ya lass, mind staying for a moment?” she nodded, sheathing her weapons as he turned back to the rest of the new recruits.
She would roll with this but damn, (Y/N) glanced quickly over at Dwalin as she raised her hand in salute and stayed that way.
“Aye lass, what is it?” Dwalin was fairly friendly at least when he wasn’t grumbling about something.
“Oh I’ve known where those two trouble makers have been and where they will be for the next few days as well” Dwalin chuckled.
“Aye, you and the rest of us” came Kili’s response as he chuckled out “would have loved to see you get knocked on your ass at least once, Dwalin”.
“Dwalin” came Fili’s voice “Never thought I would see you get your ass handed to you by a woman no less, just wait till I tell Thorin” Fili’s laughter boomed across the training area.
“Be here tomorrow at dawn” Dwalin was finally able to get out, with that he clapped Fili on the shoulder and guided him out of the training area leaving (Y/N) to herself.

Table 10: Sub-document of PAN-26 Hard train set problem 11. Transition is harder to observe with the fluency of the story.

NERQual: Evaluating the Robustness of Named Entity Recognition Models to Data Quality Issues

Denis Trautner **Philippe Thomas**

German Research Center for Artificial Intelligence (DFKI)
Speech and Language Technology Lab
{denis.trautner, philippe.thomas}@dfki.de

Abstract

Transformer-based models for Named Entity Recognition (NER) achieve strong performance on clean benchmarks, yet their reliability under degraded input in real-world settings remains insufficiently understood. In practical applications, NER systems may encounter text affected by optical character recognition, automatic speech recognition, or user-generated content. We address this gap by deriving a taxonomy of data quality perturbations from a systematic literature review, consolidating fragmented prior work into a unified framework. Guided by this taxonomy, we conduct controlled stress tests evaluating six widely used transformer-based models under varying perturbation types, intensities, and training configurations. Our results show that syntactic perturbations cause the strongest performance degradation across models in this setting, while character-level processing and positional modeling are associated with different robustness patterns. Furthermore, we show that targeted training with perturbed data can recover more than half of the lost performance with minimal impact on clean-data accuracy.

1 Introduction

Named Entity Recognition (NER) is a fundamental task in Natural Language Processing (NLP) and a cornerstone of downstream applications such as information extraction and semantic search (Yadav and Bethard, 2018; Rogers et al., 2020). Transformer-based models have set strong performance standards for NER (Devlin et al., 2019; Akbik et al., 2018), especially on established benchmarks such as CoNLL-2003 (Tjong Kim Sang and de Meulder, 2003). However, such controlled benchmark settings offer only a narrow view of model reliability under degraded real-world input (Ribeiro et al., 2020; Belinkov and Bisk, 2018).

In production environments, NER systems may encounter noisy text originating from Optical Character Recognition (OCR), Automatic Speech Recognition (ASR), or user-generated content, which can introduce spelling errors, syntactic irregularities, semantic distortions, and annotation inconsistencies (Wang et al., 2021; Belinkov and Bisk, 2018; Hamdi et al., 2023). Such data quality issues can cause severe performance degradation, with reported drops of up to one third compared to cleaner benchmark conditions (Dirkson et al., 2022; Bhadauria et al., 2024).

While robustness has attracted increasing attention in NER and related NLP tasks (Belinkov and Bisk, 2018; Lin et al., 2021; Moradi and Samwald, 2021; Bhadauria et al., 2024), existing studies typically address only limited aspects of the problem. Perturbation intensity is often operationalized and measured in different ways across studies (Li et al., 2020; Simoncini and Spanakis, 2021; Das and Paik, 2022), which renders results difficult to compare. Furthermore, previous analyses commonly focus on a restricted set of perturbation types, model families, or experimental objectives (Li et al., 2020; Bhadauria et al., 2024; Das and Paik, 2022; Simoncini and Spanakis, 2021; Lin et al., 2021; Moradi and Samwald, 2021). This fragmentation is further amplified by a lack of uniformity in evaluation settings, with studies focusing either on test-time robustness or robustness via training-time perturbations, rarely evaluating both within a single framework.

To address these limitations, we conduct a systematic experimental study that uses controlled perturbations as diagnostic probes to examine how different types and intensities of data degradation affect transformer-based NER architectures under different perturbation settings, including perturbations applied during training, at inference time, and in combination. By evaluating multiple architectures within a unified framework, our experiments enable direct comparison across perturbation types, intensity levels, and perturbation settings, while

Proceedings of the 11th Workshop on Natural User-generated Text, pages 195–209
October 28, 2026 ©2026 Association for Computational Linguistics

remaining empirically limited to English CoNLL-2003. To support further research and model auditing, we provide the dataset-agnostic perturbation framework as an open-source library, designed for easy integration with other NER datasets and reproducible evaluation.[1]

2 Related Work and Taxonomy

The robustness of NLP models under perturbed input has been studied through adversarial evaluation, data augmentation, challenge-set construction, and naturally occurring noise. Neural sequence labeling models are sensitive to small input perturbations, such as character-level noise or lexical substitutions (Belinkov and Bisk, 2018; Ebrahimi et al., 2018), and these issues persist for transformer-based architectures (Dirkson et al., 2022; Náplava et al., 2021; Lin et al., 2021). Although toolkits such as TextAttack, NLPAug, OpenAttack, and SeqAttack have made perturbation-based evaluation more accessible (Morris et al., 2020; Ma, 2019; Zeng et al., 2021; Simoncini and Spanakis, 2021), existing studies typically examine only limited subsets of perturbations (Simoncini and Spanakis, 2021; Li et al., 2020; Moradi and Samwald, 2021), making comparisons across experimental settings and model architectures difficult.

To obtain a more comprehensive picture of how data quality can be degraded in NER, we conducted a systematic literature review. From an initial pool of 400 publications, we identified 68 relevant studies on robustness and data quality in NLP and NER, including adversarial attacks, robustness evaluations, data augmentation, dataset analysis, and reported limitations. The systematic literature review was conducted following the guidelines proposed by Kitchenham and Charters (2007). Further methodological details are provided in Appendix A.

Our analysis revealed that several studies adopt a higher level of abstraction, grouping perturbations by their underlying linguistic characteristics rather than specific implementation details (Goyal et al., 2023; Bodapati et al., 2019; Simoncini and Spanakis, 2021; Qiao et al., 2024; Morris et al., 2020; Li et al., 2019). Following this paradigm, we categorize perturbations according to the linguistic level they affect: the orthographic surface form, the syntactic structure, or the semantic content. This hierarchical organization is summarized in Table 1.

[1] https://github.com/DFKI-NLP/NERQual

Perturbation type	Key References
Orthographic	
General spelling	(Ai et al., 2024; Al Sharou et al., 2021; Aliakbarzadeh et al., 2024; Bagla et al., 2021; Bhadauria et al., 2024; Chen et al., 2024; Esmaail et al., 2024; Li et al., 2020; Malykh, 2019; Malykh and Lyalin, 2018; Moradi and Samwald, 2021; Moradi et al., 2021; Namysl et al., 2020; Náplava et al., 2021; Nguyen and Chen, 2023; Qiao et al., 2024)
Keystroke errors	(Araujo et al., 2020; Bagla et al., 2023, 2021; Moradi et al., 2021)
Phonetic errors	(Nguyen and Chen, 2023; Qiao et al., 2024)
Morphological	(Náplava et al., 2021)
Elongation & char repetition	(Al Sharou et al., 2021; Moradi and Samwald, 2021; Moradi et al., 2021)
OCR errors	(Ai et al., 2024; Bhadauria et al., 2024; Boros et al., 2020; Das and Paik, 2022; Ehrmann et al., 2024; Hamdi et al., 2023; Namysl et al., 2020)
Capitalization	(Ahmed et al., 2024; Al Sharou et al., 2021; Bodapati et al., 2019; Das and Paik, 2022; Esmaail et al., 2024; Moradi and Samwald, 2021; Náplava et al., 2021; Zhu et al., 2025)
Character manipulation	(Araujo et al., 2020; Ebrahimi et al., 2018; Garg and Ramakrishnan, 2020; Goyal et al., 2023; Li et al., 2020; Nguyen and Chen, 2023)
Diacritic errors	(Al Sharou et al., 2021; Náplava et al., 2021)
Syntactic	
Grammatical	(Al Sharou et al., 2021; Chen et al., 2024; Malykh, 2019; Malykh and Lyalin, 2018; Moradi and Samwald, 2021; Qiao et al., 2024; Srinivasan and Vajjala, 2023)
Word order	(Liang et al., 2024; Moradi and Samwald, 2021; Moradi et al., 2021; Náplava et al., 2021)
Sentence struct.	(Bhadauria et al., 2024; Garg and Ramakrishnan, 2020; Li et al., 2020; Shi et al., 2024; Yuan et al., 2019; Zhu et al., 2025)
Tokenization & Segmentation	(Bernier-Colborne and Vajjala, 2024; Boros et al., 2020; Hamdi et al., 2023; Szymański et al., 2023)
Punctuation	(Al Sharou et al., 2021; Esmaail et al., 2024; Goyal et al., 2023; Náplava et al., 2021)
Semantic	
Entity replacement	(Dirkson et al., 2022; Lin et al., 2021; Lothritz et al., 2023; Morris et al., 2020; Qiao et al., 2024; Srinivasan and Vajjala, 2023; Wang and Li, 2024; Yan et al., 2022)
Context word replacement	(Araujo et al., 2020; Chen et al., 2024; Das and Paik, 2022; Dirkson et al., 2022; Garg and Ramakrishnan, 2020; Jia et al., 2019; Jin et al., 2021; Li et al., 2020; Lothritz et al., 2023; Moradi and Samwald, 2021; Morris et al., 2020; Qiang et al., 2024; Wang and Li, 2024; Wu et al., 2024; Ying et al., 2022; Zeng et al., 2021; Zhu et al., 2025)
Paraphrasing	(Goyal et al., 2023; Qiang et al., 2024; Srinivasan and Vajjala, 2023; Wang et al., 2021; Zeng et al., 2021)

Table 1: Perturbation taxonomy derived from the systematic literature review, grouping text-level perturbations into orthographic, syntactic, and semantic categories with representative perturbation types and corresponding studies. The taxonomy structures the perturbation design used in the experimental evaluation.

As illustrated in Table 1, **orthographic perturbations** are the most documented category, encompassing deviations in spelling and visual representation that do not necessarily alter grammar or meaning. These include general typos (Esmaail et al., 2024; Malykh and Lyalin, 2018; Náplava et al., 2021), keyboard errors (Araujo et al., 2020; Bagla et al., 2023; Moradi et al., 2021), and phonetic misspellings (Nguyen and Chen, 2023; Qiao et al., 2024). Technical noise often arises from OCR errors, such as misidentifying 'I' as 'l' (Hamdi et al., 2023; Namysl et al., 2020), or diacritic inconsistencies (Al Sharou et al., 2021; Náplava et al., 2021), while regional and historical variants further challenge models (Boros et al., 2020). Capitalization issues range from inconsistent casing to full "Case Ablation" (Bodapati et al., 2019; Das and Paik, 2022). Additionally, stylistic elongations (e.g., "soooo nice") and targeted character manipulations like insertion or swapping are frequently used for synthetic robustness testing (Ebrahimi et al., 2018; Goyal et al., 2023; Li et al., 2020; Moradi and Samwald, 2021).

Syntactic perturbations concern sentence structure and grammatical correctness, testing a model's ability to process structural dependencies. Disruptions include general grammatical errors (Chen et al., 2024; Malykh and Lyalin, 2018) and word order permutations (Liang et al., 2024; Náplava et al., 2021). Structural changes, such as shortening, expansion, or word merging, further challenge contextual inference (Bhadauria et al., 2024; Li et al., 2020; Shi et al., 2024). Particularly critical for NER are tokenization and segmentation errors (Ehrmann et al., 2024; Hamdi et al., 2023; Szymański et al., 2023), where incorrect splits or lost sentence boundaries hinder entity identification and disrupt the transformer's contextual window.

Semantic perturbations disrupt meaning while maintaining a plausible surface form. A common method is replacing context words with synonyms to determine if a model recognizes an entity when familiar lexical cues change (Garg and Ramakrishnan, 2020; Li et al., 2020; Morris et al., 2020; Qiang et al., 2024). More intensive methods involve entity replacement, swapping names with others from the same category (Dirkson et al., 2022) or using distributionally mismatched terms (Lin et al., 2021; Ying et al., 2022). At the sentence level, paraphrasing is used to alter surface structure while preserving underlying semantics (Srinivasan and Vajjala, 2023; Wang et al., 2021).

3 Experimental Setup

To evaluate model robustness, we follow the established methodology of constructing challenge datasets that target specific linguistic phenomena (Belinkov and Glass, 2019). By generating multiple challenge datasets through controlled variations, we compare reference performance against targeted perturbation conditions. These perturbations are intended as diagnostic probes rather than faithful simulations of naturally occurring OCR, ASR, or user-generated noise distributions. In contrast to real-world datasets that often contain multiple, interacting sources of noise, this design allows us to isolate individual degradation types and systematically vary their intensity. Following the taxonomy introduced in Section 2, we implement a suite of orthographic, syntactic, and semantic perturbations.

To quantify how different transformer architectures adapt to these degradations, we evaluate performance across four training and testing configurations:

- **Clean-Clean:** A baseline setting where models are trained, validated, and evaluated on the original, unmodified data to establish reference performance under ideal conditions
- **Clean-Perturbed:** Models are trained and validated on clean data but evaluated on a perturbed test set, allowing measurement of immediate sensitivity to data degradation at inference time
- **Perturbed-Perturbed:** Matching perturbation types are applied during training, validation, and testing. This assesses to which extent training with degraded data can mitigate performance losses
- **Perturbed–Clean:** Models are trained and validated on perturbed data and evaluated on a clean test set to assess the model's ability to generalize to unperturbed inputs

By evaluating multiple architectures within this unified framework, our approach enables a direct comparison across different perturbation types and evaluation settings while ensuring that observed performance changes are attributable to specific degradation factors.

3.1 Data

All experiments are conducted on the CoNLL-2003 English dataset (Tjong Kim Sang and de Meulder, 2003), a widely adopted benchmark for NER. Its broad use makes it well-suited for controlled robustness experiments, as it enables direct comparison between reference and perturbed conditions while keeping the results relatable to prior NER research. At the same time, CoNLL-2003 is not free of annotation inconsistencies, and cleaned variants have been proposed in prior work (Wang et al., 2019; Rücker and Akbik, 2023). We therefore treat it as a standard reference benchmark rather than as noise-free ground truth. Our framework is implemented using the Hugging Face Datasets API (HuggingFace, 2025). We utilize the standard dataset splits to ensure reproducibility and comparability with prior work. Although this study focuses on CoNLL-2003, the framework is dataset-agnostic and can be readily applied to other NER datasets supported by the same API without changes to the perturbation or evaluation pipeline.

3.2 Models

To investigate robustness across established NER architectures, we evaluate six transformer-based models that differ in architectural design, pretraining scheme, capacity, and input representation. We use pre-trained versions of these models and fine-tune them for NER on the CoNLL-2003 dataset, reflecting a common deployment pattern for encoder-based NER systems. The selection includes:

BERT (Devlin et al., 2019) serves as the baseline architecture. We use BERT-base, which consists of 12 transformer encoder layers with 12 self-attention heads and approximately 110M parameters. The model is pretrained using masked language modeling and next sentence prediction and relies on WordPiece subword tokenization.

RoBERTa (Liu et al., 2019) follows the same encoder architecture as BERT-base (12 layers, 12 heads) but is pretrained on substantially larger and more diverse text corpora. In addition, it removes the next sentence prediction objective and applies dynamic masking, resulting in a model with approximately 125M parameters. This design isolates the effects of large-scale pretraining from architectural changes.

XLNet (Yang et al., 2019) replaces masked language modeling with permutation-based language modeling to capture bidirectional context without masking. Comparable in size to BERT-base (12 layers, 12 heads, 110M parameters), it employs a two-stream self-attention mechanism and relative positional encodings.

DeBERTa (He et al., 2021) extends the BERT architecture by disentangling content and positional information within the attention mechanism. We use DeBERTa-base, which maintains a similar depth and attention configuration (12 layers, 12 heads) but increases model capacity to approximately 139M parameters.

DistilBERT (Sanh et al., 2019) is a compressed variant of BERT trained via knowledge distillation. It reduces the number of encoder layers from twelve to six, resulting in approximately 66M parameters while retaining the same hidden dimensionality and attention layout.

CANINE (Clark et al., 2022) differs fundamentally from the other models by operating directly on character-level input without explicit tokenization. It combines convolutional downsampling with Transformer layers to process long character sequences and contains approximately 104M parameters.

3.3 Perturbation Design and Implementation

To ensure a systematic and reproducible evaluation of data quality, we developed a custom perturbation framework. Our review of existing perturbation libraries such as TextAttack (Morris et al., 2020) and NLPAug (Ma, 2019) showed that they are not specifically designed for token-level NER and do not natively preserve BIO labels when token boundaries change. Furthermore, SeqAttack (Simoncini and Spanakis, 2021) does not implement the full range of perturbations identified in our taxonomy and lacks support for character-based models like CANINE (Clark et al., 2022). We therefore developed a dataset-agnostic NER perturbation framework that preserves BIO integrity through automated label re-alignment and remains architecture-neutral across subword and character-level models.

The profile-based framework organizes perturbations into orthographic, semantic, and syntactic classes. Each class can be configured to execute specific methods or the entire category. A dedicated control method manages candidate selection and distributes selected tokens among the active perturbation methods while enforcing selection constraints. A detailed overview of all implemented methods and examples is available in Appendix B.

We initialize the random number generators of

Python, NumPy, PyTorch, and CUDA with a shared global seed to improve reproducibility. To assess variability across runs and hyperparameter settings, each experiment is executed across five different seeds and three distinct learning rates (1e-5, 3e-5, 5e-5). Furthermore, our framework incorporates entity-protection, allowing us to isolate perturbations to either the context or the entities themselves. To create controlled stress-test conditions with increasing perturbation intensity, perturbations are applied at three levels of 10%, 20%, and 30%. This combination of variables results in a total of 2,700 training runs. While the framework is a custom development, the individual transformation methods are inspired by and adapted from established literature. Within the semantic category, we implement three primary methods for lexical and contextual replacement. Following Chen et al. (2024); Srinivasan and Vajjala (2023), we use WordNet-based synonyms and antonyms to introduce lexical variation. Inspired by Li et al. (2021) and Li et al. (2020), we use static GloVe (Pennington et al., 2014) embeddings to identify semantically similar neighbors for replacement. Finally, drawing on Lin et al. (2021) and Ma (2019), we employ Masked Language Modeling (MLM) via an independent ALBERT (Lan et al., 2020) model to generate contextually fitting perturbations.

4 Results

This section presents the experimental findings across all four evaluation settings. Performance is assessed using the entity-level micro-F1 score, necessitating an exact match for both the entity boundary and the predicted category.

4.1 Baseline

The `Clean-Clean` setting serves as performance ceiling, establishing a reference for model behavior under ideal conditions. As summarized in Table 2, RoBERTa and DeBERTa achieve the highest scores, followed by BERT and XLNet. DistilBERT and CANINE show slightly lower results. The low standard deviations across all models confirm the stability of the fine-tuning process.

4.2 Perturbed Test Data

This section evaluates model sensitivity to inference-time noise by testing architectures trained on clean data against perturbed test sets. As shown in Figure 1, performance correlates negatively with the proportion of modified tokens. Syntactic perturbations cause the most severe degradation, dropping the mean F1-score below 76% at 30% intensity. This category also exhibits the highest volatility with a standard deviation of $\sigma = \pm 1.5$ at 30%. In contrast, orthographic modifications are best compensated by the models, remaining above 80% at the same noise level with lower volatility ($\sigma = \pm 0.9$). Semantic perturbations occupy an intermediate position, with a slowing degradation rate that leads semantic and orthographic performance to converge at maximum intensity.

Model	Micro-F1 (%)	Std. Dev. σ
BERT	91.8	$\pm$ 0.3
RoBERTa	92.9	$\pm$ 0.2
DeBERTa	92.7	$\pm$ 0.3
DistilBERT	90.3	$\pm$ 0.4
CANINE	89.4	$\pm$ 0.5
XLNet	91.8	$\pm$ 0.3

Table 2: Baseline Performance on Clean CoNLL-2003 Test Data (clean-clean setting).

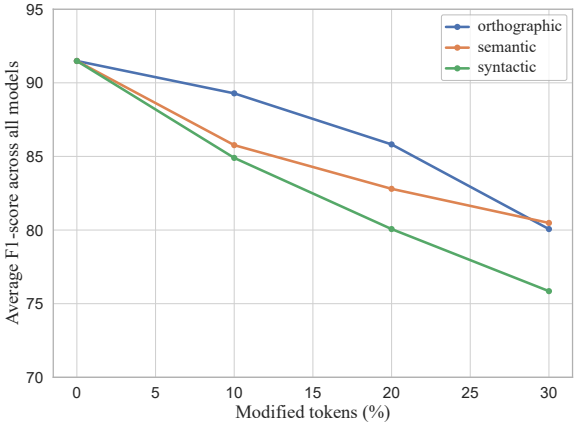


Figure 1: Mean Micro-F1 across models under increasing levels of orthographic, semantic, and syntactic test-set perturbations.

The progression of performance degradation is non-linear. Syntactic disturbances show the largest initial drop at 10% perturbation intensity. Conversely, orthographic perturbations follow a flatter trajectory initially, with a stronger decline between 20% and 30%. This pattern suggests that local surface-form changes remain easier to absorb at lower intensities, while structural perturbations affect model predictions earlier.

Detailed results per model are reported in Table 3. RoBERTa and DeBERTa achieve the strongest performance across all perturbation types, maintaining F1 scores above 85% at the initial perturbation level of 10%. This stability is reflected in

Type	Noise	BERT	RoBERTa	DeBERTa	DistilBERT	CANINE	XLNet
Baseline	0%	91.8 ± 0.3	**92.9** ± 0.2	92.7 ± 0.3	90.3 ± 0.4	89.4 ± 0.5	91.8 ± 0.3
Orthographic	10%	89.7 ± 0.6	**91.1** ± 0.5	90.7 ± 0.5	86.0 ± 0.8	88.6 ± 0.4	89.6 ± 0.7
	20%	85.9 ± 0.9	**88.0** ± 0.6	86.9 ± 0.7	81.3 ± 1.2	86.9 ± 0.4	86.3 ± 0.8
	30%	79.8 ± 1.2	**82.7** ± 0.9	82.2 ± 1.0	76.1 ± 1.6	**83.9** ± 0.6	80.0 ± 1.2
Semantic	10%	86.1 ± 0.8	**88.0** ± 0.7	87.7 ± 0.7	83.2 ± 1.0	83.4 ± 0.9	86.2 ± 0.8
	20%	83.3 ± 1.0	**85.4** ± 0.8	84.8 ± 0.9	78.5 ± 1.2	80.9 ± 1.1	83.9 ± 1.0
	30%	81.3 ± 1.2	83.6 ± 1.0	**83.7** ± 1.1	74.9 ± 1.5	78.1 ± 1.3	81.3 ± 1.2
Syntactic	10%	84.9 ± 1.0	86.7 ± 0.8	**86.9** ± 0.9	81.8 ± 1.3	82.7 ± 1.1	86.4 ± 1.0
	20%	79.9 ± 1.3	82.1 ± 1.0	**82.7** ± 1.1	76.6 ± 1.6	77.8 ± 1.3	81.3 ± 1.3
	30%	76.1 ± 1.5	77.7 ± 1.2	**78.8** ± 1.2	72.2 ± 2.2	72.6 ± 1.6	77.7 ± 1.5

Table 3: Model performance with a perturbed test dataset (entity-level Micro-F1, %) $\pm\sigma$ over five random seeds and three different learning rates on CoNLL-2003 (clean-perturbed setting).

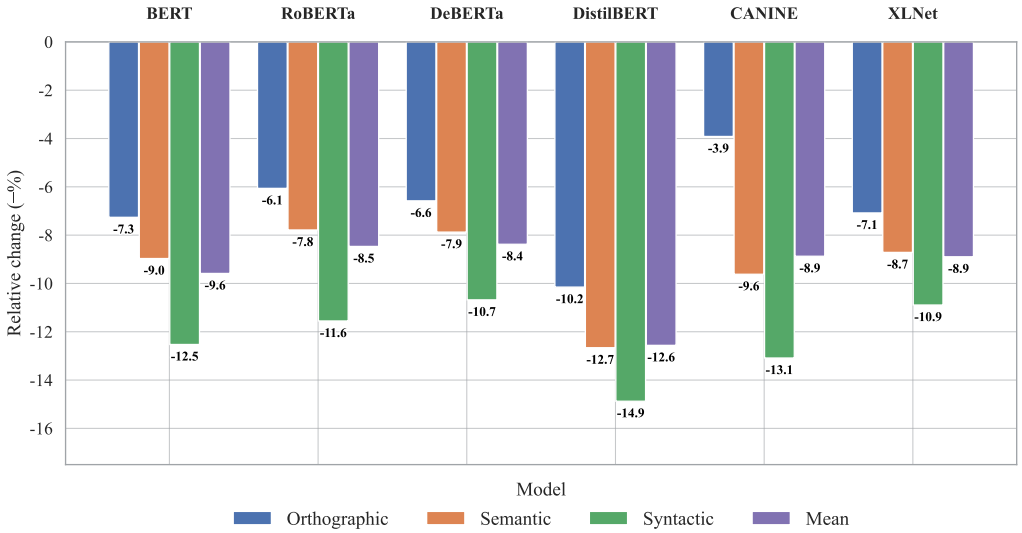


Figure 2: Relative performance change per model averaged across all intensity levels and compared to clean baselines. These averages may obscure significant failures at higher intensities; for example, stability at 10% orthographic perturbation partially offsets the sharp performance degradation seen at 30%.

their lower standard deviation ($\sigma \approx 0.8$) compared to the model average ($\sigma \approx 1.2$), indicating more stable behavior under these perturbation probes.

Model-specific robustness patterns become evident under specific degradation types. CANINE shows strong robustness to orthographic perturbations: despite a lower clean-data baseline (89.4%), it achieves the highest performance at the maximum perturbation intensity, reaching 83.9% F1 while maintaining low variability ($\sigma = 0.6$). This pattern is consistent with character-level processing being less affected by typo-induced subword segmentation errors. As illustrated in Figure 2, this corresponds to a relative drop of -3.9% for CANINE. However, this advantage is category-specific: under semantic (-9.6%) and syntactic (-13.1%) perturbations, CANINE exhibits a higher relative decline than the strongest models.

Syntactic perturbations remain the most challenging category, with even the strongest architectures dropping below 79% F1 at maximum intensity. DeBERTa achieves the highest syntactic score (78.8%), which may be related to its disentangled attention mechanism, although this interpretation cannot be isolated from differences in pretraining and capacity. Figure 2 also shows that DistilBERT has the largest average loss (-12.6%), indicating that parameter distillation may reduce the redundancy required to reconstruct contextual information from sequences where structural or semantic cues are fragmented.

4.3 Perturbed Training & Test Data

This section examines whether incorporating perturbations during training mitigates the performance degradation identified in Section 4.2.

Type	Noise	BERT	RoBERTa	DeBERTa	DistilBERT	CANINE	XLNet
Baseline	0%	91.8 ± 0.3	**92.9** ± 0.2	92.7 ± 0.3	90.3 ± 0.4	89.4 ± 0.5	91.8 ± 0.3
Orthographic	10%	90.1 ± 0.4	91.6 ± 0.3	**91.7** ± 0.4	87.8 ± 0.6	88.7 ± 0.5	90.7 ± 0.5
	20%	88.7 ± 0.6	**90.1** ± 0.5	90.0 ± 0.5	86.5 ± 0.8	87.9 ± 0.5	89.2 ± 0.6
	30%	86.4 ± 0.8	**89.0** ± 0.8	**89.0** ± 0.7	84.1 ± 1.2	86.9 ± 0.6	86.6 ± 0.6
Semantic	10%	89.6 ± 0.6	90.3 ± 0.5	**91.0** ± 0.5	87.7 ± 0.8	87.6 ± 0.7	89.9 ± 0.5
	20%	86.7 ± 0.8	88.3 ± 0.7	**89.1** ± 0.7	84.6 ± 1.0	84.5 ± 0.9	86.9 ± 0.7
	30%	85.3 ± 0.9	87.5 ± 0.8	**88.4** ± 0.7	81.9 ± 1.1	82.2 ± 1.0	85.2 ± 0.9
Syntactic	10%	88.4 ± 0.7	90.0 ± 0.6	**90.4** ± 0.6	86.6 ± 0.9	85.8 ± 0.9	89.0 ± 0.8
	20%	85.3 ± 0.9	88.6 ± 0.8	**89.2** ± 0.7	83.2 ± 1.2	82.5 ± 1.1	87.7 ± 1.0
	30%	83.2 ± 1.1	86.0 ± 0.9	**87.5** ± 0.9	80.2 ± 1.5	80.3 ± 1.2	86.2 ± 1.0

Table 4: Model performance with perturbed training, validation, and test dataset (entity-level Micro-F1, %) $\pm\sigma$ over five random seeds and three different learning rates on CoNLL-2003 (perturbed-perturbed setting).

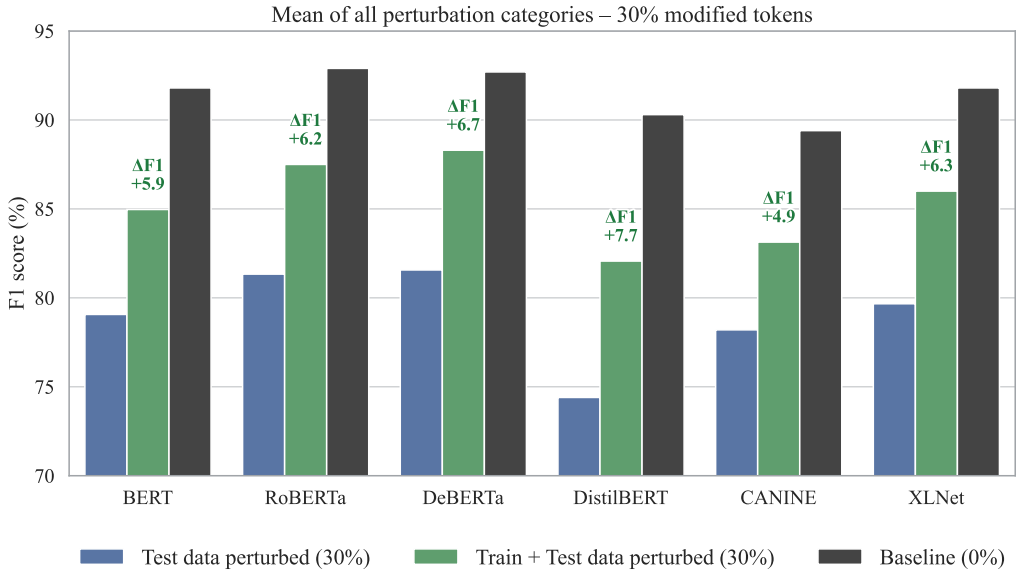


Figure 3: Mean Micro-F1 scores across all perturbation categories at 30% modified tokens, comparing clean data, perturbed test data, and perturbed training–test data for each model. ΔF1 denotes the absolute improvement in Micro-F1 (in percentage points) achieved by training with perturbed data compared to perturbing the test set only.

For all architectures, Table 4 shows a noticeable performance recovery. The degradation curves follow a flatter trajectory than in the clean-perturbed setting, suggesting that training with degraded data improves stability across perturbation intensities. Average standard deviations decrease from $\sigma \approx 1.2$ to $\sigma \approx 0.8$.

To quantify this recovery, we examine the maximum load limit at 30% modified tokens in Figure 3. At this intensity, models recover on average 58.1% of the initial performance loss for orthographic, 52.1% for syntactic, and 41.9% for semantic perturbations.

The recovery differs across models and perturbation categories. At the category-mean level shown in Figure 3, DeBERTa and XLNet show high absolute gains of +6.7 and +6.3 percentage points (pp), respectively, compared to the clean-perturbed setting. These gains indicate that both models benefit substantially from exposure to perturbed examples during training. Their strong recovery under syntactic perturbations is also consistent with the view that positional modeling mechanisms may help under structural degradation. However, this effect cannot be isolated entirely.

DistilBERT records the highest absolute gain of +7.7 pp, indicating substantial improvement when the model is trained on data from the same perturbation distribution used at test time. This result should not be interpreted as evidence that compression itself improves adaptability. Rather, it shows that a model with larger clean-perturbed losses can also exhibit larger recoverable margins under perturbation-aware training. Conversely, CANINE

shows a lower gain of +4.9 pp, which is consistent with its stronger clean-perturbed performance under orthographic perturbations and therefore less room for recovery in that category.

DeBERTa also shows the most consistent robustness gains after perturbed training. While DeBERTa and RoBERTa perform similarly on clean data, DeBERTa achieves higher gains across all perturbation categories, including +4.7 pp in the semantic domain compared to +3.9 pp for RoBERTa. This recurring advantage suggests that DeBERTa's separation of content and positional information may support adaptation to perturbed inputs, although the present experiments do not isolate this mechanism from other model differences.

Despite this adaptability, absolute F1-scores for syntactic perturbations remain below those observed for orthographic perturbations after perturbed training. This indicates that disrupted sentence structure remains harder to handle than local spelling variation, even when the corresponding perturbation type is seen during training.

4.4 Perturbed Training Data – Clean Test Data

This section examines how training on perturbed data affects model performance on the unmodified test set. Evaluating on clean test data allows us to assess whether perturbation-aware training reduces reference performance when no input degradation is present. Comprehensive results are documented in Appendix C.

All architectures remain comparatively stable under training-phase perturbations. Unlike the larger declines observed during perturbed inference, even a 30% perturbation rate during training results in limited performance loss, with standard deviations remaining close to baseline levels. Orthographic changes have the least impact. RoBERTa remains nearly stable at 92.6% (-0.3 pp), while DeBERTa marginally exceeds its baseline with 92.8% (+0.1 pp) at the 20% level. This suggests that moderate orthographic variance may function as data augmentation, potentially preventing overfitting and improving model robustness.

Semantic and syntactic perturbations lead to slightly larger declines, but the overall losses remain limited. Even at 30% syntactic modification, RoBERTa and DeBERTa only drop to 90.5% (-2.4 pp) and 91.0% (-1.7 pp), respectively.

These results show that perturbation-aware training can improve robustness under degraded test conditions while largely preserving performance on unmodified inputs. This suggests that, within our controlled CoNLL-2003 setting, the robustness gains observed in Section 4.3. do not come at the cost of substantial clean-data degradation.

5 Conclusion

In this paper, we evaluated the robustness of six transformer architectures for NER using a newly developed, NER-specific perturbation framework. The framework is grounded in a structured taxonomy of perturbation types and enables controlled, reproducible evaluation of data quality degradation across different intensities and training settings.

Within our English CoNLL-2003 setting, the results show clear differences in how architectures respond to controlled perturbation probes. Syntactic perturbations cause the strongest performance degradation overall, while DeBERTa and XLNet exhibit the smallest average relative declines under syntactic perturbations. This pattern is consistent with the relevance of positional modeling, although the present experiments do not isolate this factor from other model differences. Across perturbation types, RoBERTa and DeBERTa show the strongest overall stability as the proportion of modified tokens increases. CANINE shows strong robustness to orthographic perturbations, which is consistent with its character-level input representation, while DistilBERT shows larger performance drops under the evaluated perturbation conditions.

Furthermore, we show that perturbation-aware training substantially mitigates performance degradation at inference time. This strategy recovers much of the lost performance while largely preserving performance on unmodified inputs. The results therefore indicate that targeted perturbation-aware training can improve robustness in controlled stress-test settings without requiring changes to the underlying model architecture.

Future research should extend this analysis to additional datasets, domains, languages, and model classes to assess the generalizability of the observed patterns. Incorporating empirically observed real-world error distributions and validating perturbation plausibility would further clarify how controlled probes relate to naturally occurring OCR, ASR, and user-generated noise. Overall, the presented framework and findings provide a reusable basis for auditing NER robustness under imperfect input conditions.

Limitations

This study is subject to several limitations that should be considered when interpreting the results.

First, while the perturbation taxonomy is grounded in prior work, it may not capture the full spectrum of data quality issues encountered in practice. Although the taxonomy was derived from a broad literature base, certain perturbation types or variants may not be covered, which may limit the completeness of the robustness analysis.

Second, the experimental evaluation is restricted to six established encoder-based transformer architectures. While these models represent widely used and diverse design choices for token-level NER, the findings may not directly generalize to generative LLMs, multilingual models, domain-specific pretrained models, or more recent architectural variants.

Third, all experiments are conducted on the English CoNLL-2003 dataset. This benchmark enables comparability with prior NER work, but it consists of relatively homogeneous newswire text and contains documented annotation inconsistencies. Robustness patterns may differ in cleaned CoNLL variants, additional languages, or domain-specific and informal settings such as medical, legal, or social media text.

Finally, the applied perturbations are synthetically generated and uniformly distributed. They should therefore be interpreted as controlled robustness probes rather than faithful simulations of naturally occurring OCR, ASR, or user-generated noise. While the perturbation methods are linguistically motivated and grounded in prior work, we do not include a human evaluation of their realism or plausibility. Human validation and empirically observed error distributions could further clarify how controlled perturbations relate to real-world degradation.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive feedback and helpful suggestions. This work originated from Denis Trautner's master's thesis, supervised by Philippe Thomas and Alexander Dück. We are grateful to Alexander Dück for his academic guidance and valuable feedback throughout the thesis.

References

Anil Ahmed, Degen Huang, Syed Yasser Arafat, and Imran Hameed. 2024. Enriching urdu ner with bert embedding, data augmentation, and hybrid encoder-cnn architecture. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(4):1–38.

Chaoyi Ai, Yong Jiang, Shen Huang, Pengjun Xie, and Kewei Tu. 2024. Learning robust named entity recognizers from noisy data with retrieval augmentation. *Preprint*, arXiv:2407.18562.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Khetam Al Sharou, Zhenhao Li, and Lucia Specia. 2021. Towards a better understanding of noise in natural language processing. *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62.

Amirhossein Aliakbarzadeh, Lucie Flek, and Akbar Karimi. 2024. Exploring robustness of multilingual LLMs on real-world noisy data. In *Eighth Widening NLP Workshop (WiNLP 2024) Phase II*.

Vladimir Araujo, Andres Carvallo, Carlos Aspillaga, and Denis Parra. 2020. On adversarial examples for biomedical nlp tasks. *Preprint*, arXiv:2004.11157.

Kartikay Bagla, Shivam Gupta, Ankit Kumar, and Anuj Gupta. 2023. Noisy text data: foible of popular transformer based nlp models. In *The Third International Conference on Artificial Intelligence and Machine Learning Systems*, pages 1–5, New York, NY, USA. ACM.

Kartikay Bagla, Ankit Kumar, Shivam Gupta, and Anuj Gupta. 2021. Noisy text data: Achilles' heel of popular transformer based nlp models. *Preprint*, arXiv:2110.03353.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Gabriel Bernier-Colborne and Sowmya Vajjala. 2024. Annotation errors and ner: A study with ontonotes 5.0. *Preprint*, arXiv:2406.19172.

Divya Bhadauria, Alejandro Sierra-Múnera, and Ralf Krestel. 2024. The effects of data quality on named entity recognition. *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 79–88.

Sravan Bodapati, Hyokun Yun, and Yaser Al-Onaizan. 2019. Robustness to capitalization errors in named entity recognition. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 237–242, Hong Kong, China. Association for Computational Linguistics.

Emanuela Boros, Ahmed Hamdi, Elvys Linhares Pontes, Luis Adrián Cabrera-Diego, Jose G. Moreno, Nicolas Sidere, and Antoine Doucet. 2020. Alleviating digitization errors in named entity recognition for historical documents. *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 431–441.

Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2024. Context-aware adversarial attack on named entity recognition. In *Proceedings of the Ninth Workshop on Noisy and User-generated Text (W-NUT 2024)*, pages 11–16, San Ġiljan, Malta. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine : Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Sudeshna Das and Jiaul Paik. 2022. Resilience of named entity recognition models under adversarial attack. *Proceedings of the First Workshop on Dynamic Adversarial Data Collection*, pages 1–6.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *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)*, pages 4171–4186.

Anne Dirkson, Suzan Verberne, and Wessel Kraaij. 2022. Breaking bert: Understanding its vulnerabilities for named entity recognition through adversarial attack. *Preprint*, arXiv:2109.11308.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36.

Maud Ehrmann, Ahmed Hamdi, Elvys Linhares Pontes, Matteo Romanello, and Antoine Doucet. 2024. Named entity recognition and classification in historical documents: A survey. *ACM Computing Surveys*, 56(2):1–47.

Naji Esmaail, Nazlia Omar, Masnizah Mohd, Fariza Fauzi, and Zainab Mansur. 2024. Named entity recognition in user-generated text: A systematic literature review. *IEEE Access*, 12:136330–136353.

Siddhant Garg and Goutham Ramakrishnan. 2020. Bae: Bert-based adversarial examples for text classification. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181.

Shreya Goyal, Sumanth Doddapaneni, Mitesh M. Khapra, and Balaraman Ravindran. 2023. A survey of adversarial defenses and robustness in nlp. *ACM Computing Surveys*, 55(14s):1–39.

Ahmed Hamdi, Elvys Linhares Pontes, Nicolas Sidere, Mickaël Coustaty, and Antoine Doucet. 2023. In-depth analysis of the impact of ocr errors on named entity recognition and linking. *Natural Language Engineering*, 29(2):425–448.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-enhanced bert with disentangled attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

HuggingFace. 2025. Datasets - library.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. 2019. Certified robustness to adversarial word substitutions. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4127–4140.

Lifeng Jin, Linfeng Song, Kun Xu, and Dong Yu. 2021. Instance-adaptive training with noise-robust losses against noisy labels. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5647–5663.

Barbara Kitchenham and Stuart Charters. 2007. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *Eighth International Conference on Learning Representations*.

Dianqi Li, Yizhe Zhang, Hao Peng, Liqun Chen, Chris Brockett, Ming-Ting Sun, and Bill Dolan. 2021. Contextualized perturbation for textual adversarial attack. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5053–5069.

Jinfeng Li, Shouling Ji, Tianyu Du, Bo Li, and Ting Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium, NDSS 2019, San Diego, California, USA, February 24-27, 2019*. The Internet Society.

Linyang Li, Ruotian Ma, Qipeng Guo, Xiangyang Xue, and Xipeng Qiu. 2020. Bert-attack: Adversarial attack against bert using bert. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6193–6202.

Xiaobo Liang, Runze Mao, Lijun Wu, Juntao Li, Min Zhang, and Qing Li. 2024. Enhancing low-resource nlp by consistency training with data and model perturbations. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:189–199.

Bill Yuchen Lin, Wenyang Gao, Jun Yan, Ryan Moreno, and Xiang Ren. 2021. RockNER: A simple method to create adversarial examples for evaluating the robustness of named entity recognition models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3728–3737, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Du Jingfei, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Cedric Lothritz, Saad Ezzini, Christoph Purschke, Tegawendé Bissyandé, Jacques Klein, Isabella Olariu, Andrey Boytsov, Clément Lefebvre, and Anne Goujon. 2023. Comparing pre-training schemes for luxembourgish bert models. *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 17–27.

Edward Ma. 2019. Nlpaug - nlp augmentation framework.

Valentin Malykh. 2019. Robust to noise models in natural language processing tasks. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 10–16.

Valentin Malykh and Vladislav Lyalin. 2018. Named entity recognition in noisy domains. In *2018 International Conference on Artificial Intelligence Applications and Innovations (IC-AIAI)*, pages 60–65. IEEE.

Milad Moradi, Kathrin Blagec, and Matthias Samwald. 2021. Deep learning models are not robust against noise in clinical text. *Preprint*, arXiv:2108.12242.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Marcin Namysl, Sven Behnke, and Joachim Köhler. 2020. NAT: Noise-aware training for robust neural sequence labeling. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1501–1517, Online. Association for Computational Linguistics.

Jakub Náplava, Martin Popel, Milan Straka, and Jana Straková. 2021. Understanding model robustness to user-generated noisy texts. *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 340–350.

Minh Nguyen and Nancy Chen. 2023. Firo: Finite-context indexing of restricted output space for nlp models facing noisy input. *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 834–845.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. 2024. Prompt perturbation consistency learning for robust language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1357–1370, St. Julian's, Malta. Association for Computational Linguistics.

Haishan Qiao, Yuliang Zhu, Meiying San, Zeyang Chen, Sheqiang Zhao, Yan Chen, Zhangyan Li, and Shenghui Shi. 2024. A named entity recognition method for fields including electrical power based on data enhancement. In *2024 2nd International Conference on Artificial Intelligence and Power Engineering (AIPE)*, pages 11–16. IEEE.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Susanna Rücker and Alan Akbik. 2023. CleanCoNLL: A nearly noise-free named entity recognition dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8628–8645, Singapore. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

Shenghui Shi, Zhangyan Li, Lin Yin, Zhaoying Chai, Kechen Fan, Daguang Jiang, Liangfei Zheng, and Dongqi Huang. 2024. Research and data analysis on relationship extraction methods based on multi-domain texts. In *2024 9th International Conference on Big Data Analytics (ICBDA)*, pages 197–202. IEEE.

Walter Simoncini and Gerasimos Spanakis. 2021. Seqattack: On adversarial attacks for named entity recognition. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 308–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Akshay Srinivasan and Sowmya Vajjala. 2023. A multilingual evaluation of NER robustness to adversarial inputs. In *Proceedings of the 8th Workshop on Representation Learning for NLP (RepL4NLP 2023)*, pages 40–53, Toronto, Canada. Association for Computational Linguistics.

Piotr Szymański, Lukasz Augustyniak, Mikolaj Morzy, Adrian Szymczak, Krzysztof Surdyk, and Piotr Żelasko. 2023. Why aren't we ner yet? artifacts of asr errors in named entity recognition in spontaneous speech transcripts. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1761.

Erik Tjong Kim Sang and Fien de Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial GLUE: A multi-task benchmark for robustness evaluation of language models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Hongtao Wang and Ang Li. 2024. Are deep neural networks robust to named entities? an adversarial attack and defense perspective. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12556–12560. IEEE.

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. CrossWeigh: Training named entity tagger from imperfect annotations. 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)*, pages 5154–5163, Hong Kong, China. Association for Computational Linguistics.

Sihao Wu, Xiaoling Wu, Rui Xie, and Wentao Zhan. 2024. Chinese ner based on adversarial training and interactive attention. In *Proceedings of the 2024 16th International Conference on Machine Learning and Computing*, pages 557–561, New York, NY, USA. ACM.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158.

Jun Yan, Yang Xiao, Sagnik Mukherjee, Bill Yuchen Lin, Robin Jia, and Xiang Ren. 2022. On the robustness of reading comprehension models to entity renaming. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 508–520, Seattle, United States. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zheyu Ying, Jinglei Zhang, Rui Xie, Guochang Wen, Feng Xiao, Xueyang Liu, and Shikun Zhang. 2022. 3rs: Data augmentation techniques using document contexts for low-resource chinese named entity recognition. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824.

Guoyang Zeng, Fanchao Qi, Qianrui Zhou, Tingji Zhang, Zixian Ma, Bairu Hou, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. OpenAttack: An open-source textual adversarial attack toolkit. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 363–371, Online. Association for Computational Linguistics.

Jizhao Zhu, Akang Shi, Zixuan Li, Long Bai, Xiaolong Jin, Jiafeng Guo, and Xueqi Cheng. 2025. Towards robust universal information extraction: Dataset, evaluation, and solution. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28052–28070, Vienna, Austria. Association for Computational Linguistics.

A Systematic Literature Review

The systematic literature review was conducted following the guidelines of Kitchenham and Charters (2007). The goal was to identify and structure prior work on robustness and data quality issues in Natural Language Processing (NLP), with a focus on Named Entity Recognition (NER) and transformer-based models.

To ensure a consistent and transparent selection process, predefined inclusion and exclusion criteria were applied throughout the review. These criteria are summarized in Table 5.

Category	Include	Exclude
Topic	Robustness or data quality issues in NLP or NER	No clear NLP focus or no substantive robustness analysis
Model Type	Transformer-based encoder architectures	Non-transformer models
Quality Issue	Explicit and reproducible perturbations	Unspecified or opaque input changes
Publication	Peer-reviewed conference or journal papers; preprints	Books or non-scientific literature
Time Frame	2017–2025	Before 2017
Language	English or German	Other languages
Availability	Full text accessible	No full-text access
Citations	More than 10 citations	10 or fewer citations

Table 5: Inclusion and exclusion criteria used in the systematic literature review.

To refine the search strategy, a pilot search evaluated candidate keywords related to robustness and noise. High-yield terms (e.g., *noise*, *robust*, *data quality*, *adversarial*) were retained, while low-yield terms were discarded.

Searches were conducted in the ACL Anthology, IEEE Xplore, ACM Digital Library, and arXiv as these were identified as the most promising sources during the pilot search. For each database, the first 100 results ranked by relevance were screened. Studies were filtered through deduplication, title and abstract screening, and full-text assessment.

The final search query was:

```
("natural language" OR "NLP" OR "named
entity recognition" OR "NER")
AND
("data quality" OR robust OR nois* OR
perturbation OR adversarial)
```

The screening process is summarized in Figure 4.

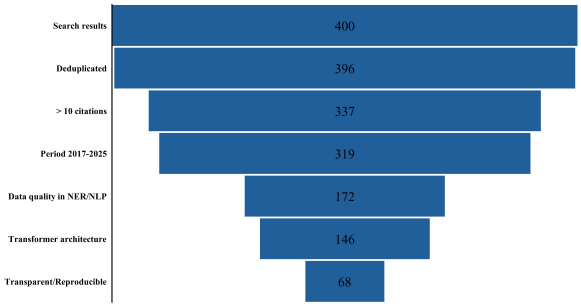


Figure 4: Study selection and screening process for the systematic literature review. Numbers indicate the remaining studies after each filtering step.

Out of the 400 initially identified studies, 68 met all inclusion criteria and were included in the final review. These studies informed the perturbation taxonomy and experimental design used in this work.

B Implemented Methods & Examples

This appendix provides qualitative examples of the perturbation methods implemented in this work, covering orthographic, syntactic, and semantic categories. The examples illustrate how controlled data quality perturbations are applied to the CoNLL-2003 corpus and correspond to the perturbation taxonomy introduced in Section 2. Orthographic, syntactic, and semantic perturbations are shown in Tables 6, 7, and 8, respectively. All perturbation implementations are publicly available in our GitHub repository `https://github.com/DFKI-NLP/NERQual` .

C Results - Perturbed Training Data

Detailed results for experiments with perturbed training and validation data evaluated on the clean test set are reported in Table 9. Across models, performance changes remain small even at higher perturbation levels.

Method	Original Text	Perturbed Text
Insertion	The company said it expects higher profits next year.	The company said it expects higbher profits next year.
Deletion	Peter Blackburn reported from London.	Peter Blackburn reported from Londo~~n~~.
Substitution	France and Germany agreed on the proposal.	France and Germany agrred on the proposal.
Case change	IBM said the market was stable after recent reports.	Ibm said the market was stable after recent reports.
Character swap	. . . would continue to support economic reforms in Europe.	. . . would continue to uspport economic reforms in Europe.
Diacritics	José Mourinho joined the club after talks with Chelsea officials.	Jose Mourinho joined the club after talks with Chelsea officials.
Homoglyphs	Google announced a new partnership with Apple on Monday.	Google announced a new partnership with App1e on Monday.

Table 6: Overview of implemented orthographic perturbation methods with examples from the CoNLL-2003 corpus before and after applying orthographic transformations that simulate typical spelling and typing errors. Modified tokens are highlighted in red.

Method	Original Text	Perturbed Text
Insert punctuation	The United Nations called for new peace talks in Geneva.	The United , Nations called for new peace talks in Geneva.
Delete punctuation	. . . from Bank of America, New York, and the United States Treasury met in London.	. . . from Bank of America ~~,~~ New York, and the United States Treasury met in London.
Remove spaces	France and Germany signed the agreement yesterday.	Franceand Germany signed the agreement yesterday.
Split words	Microsoft announced new software on Monday.	Micro␣soft announced new software on Monday.
Delete words	Tony Blair met President Bush in London.	Tony Blair met ~~President~~ Bush in London.
Repetition	IBM reported strong results in the first quarter.	IBM IBM reported strong results in the first quarter.
Swap	Google announced a new partnership with Apple.	Google announced a new partnership Apple with.

Table 7: Overview of implemented syntactic perturbation methods with examples from the CoNLL-2003 corpus before and after applying syntactic transformation operations that simulate structural deviations in sentence organization and token segmentation. Modified tokens are highlighted in red.

Method	Original Text	Perturbed Text
WordNet-based synonym substitution	The foreign ministry's Shen told Reuters Television . . .	The foreign ministry's Shen told Haaretz Video . . .
WordNet-based antonym substitution	German first-time registrations of motor vehicles jumped . . .	German unwary registrations of motor vehicles jumped . . .
Embedding-based substitution (GloVe)	The meeting was attended by Tony Blair in London.	The meeting was operated by Tony Blair in London.
Contextual substitution (MLM via ALBERT)	Talks between France and Germany continued in Brussels.	Talks between France and Germany collapsed in Brussels.

Table 8: Overview of implemented semantic perturbation methods with examples from the CoNLL-2003 corpus before and after applying WordNet-, GloVe-, and MLM-based substitution techniques (here using ALBERT) to generate semantic variations. Modified tokens are highlighted in red.

Type	Noise	BERT	RoBERTa	DeBERTa	DistilBERT	CANINE	XLNet
Baseline	0%	91.8 ± 0.3	**92.9** ± 0.2	92.7 ± 0.3	90.3 ± 0.4	89.4 ± 0.5	91.8 ± 0.3
Orthographic	10%	91.5 ± 0.4	**92.5** ± 0.3	92.6 ± 0.3	89.9 ± 0.4	89.3 ± 0.4	91.6 ± 0.4
	20%	91.3 ± 0.3	92.7 ± 0.3	**92.8** ± 0.4	89.5 ± 0.3	89.0 ± 0.5	91.6 ± 0.3
	30%	90.9 ± 0.4	92.6 ± 0.4	**92.5** ± 0.4	88.8 ± 0.4	88.8 ± 0.5	91.3 ± 0.4
Semantic	10%	90.7 ± 0.4	91.7 ± 0.3	**92.2** ± 0.4	89.5 ± 0.5	88.9 ± 0.5	91.2 ± 0.4
	20%	90.8 ± 0.4	91.8 ± 0.4	**92.3** ± 0.4	88.9 ± 0.4	89.0 ± 0.6	91.2 ± 0.4
	30%	90.1 ± 0.3	91.6 ± 0.4	**92.0** ± 0.5	88.1 ± 0.5	88.7 ± 0.6	90.8 ± 0.5
Syntactic	10%	90.1 ± 0.3	91.7 ± 0.4	**91.8** ± 0.3	88.8 ± 0.5	88.5 ± 0.4	90.1 ± 0.4
	20%	89.7 ± 0.3	91.2 ± 0.3	**91.3** ± 0.4	87.9 ± 0.5	87.8 ± 0.5	89.4 ± 0.5
	30%	89.1 ± 0.5	90.5 ± 0.4	**91.0** ± 0.4	87.7 ± 0.6	87.1 ± 0.5	88.7 ± 0.5

Table 9: Model performance with a perturbed training & validation dataset (entity-level Micro-F1, %) $\pm\sigma$ over five random seeds and three different learning rates (1e-5, 3e-5, 5e-5) on CoNLL-2003 (perturbed-clean setting).

Not All Duplicates Are Coordination: Generic vs. Non-Generic Duplicate Campaigns in Information Operations

Ashfaq Ali Shafin[1] **Khandaker Mamun Ahmed**[2]
[1] Mathematics and Computer Science, Augustana College, USA
shafinashfaqali21@gmail.com
[2] Beacom College of Computer and Cyber Sciences, Dakota State University, USA
khandakermamun.ahmed@dsu.edu

Abstract

Duplicate content is widely used to study coordinated behavior in social media information operations (IOs), but not all repetition provides equally meaningful evidence of coordination. Generic, reusable, or low-information posts may create noisy account-account links when projected into coordination graphs. We study this problem using 187,000 English-language tweets from six Russian Twitter Information Operations datasets. We introduce a generic/non-generic distinction for duplicate campaigns, label tweets using an LLM-assisted protocol with independent human validation, and train supervised classifiers over sentence embeddings to scale the labels. We construct duplicate campaigns using lexical similarity and two embedding-based methods. Generic campaigns are rare under lexical matching but account for nearly 39% of campaigns detected by embedding-based methods. Restricting graphs to non-generic campaigns reduces graph size and the largest connected component while increasing density, suggesting a smaller but more focused coordination structure. These findings show that duplicate-based coordination analysis should consider both textual similarity and semantic specificity.

1 Introduction

Social media platforms are used for political discussion, news sharing, and public mobilization, but they are also exploited by coordinated actors to amplify narratives, manipulate attention, and spread misleading or harmful content (Recabarren et al., 2023; Zannettou et al., 2019; Luceri et al., 2024; Shafin and Ahmed, 2025b,a). Detecting such behavior is difficult because researchers often lack privileged platform-side metadata, deleted account histories, or complete interaction traces. As a result, public content-based signals, including repeated or near-repeated text, remain important for studying possible coordination (Pacheco et al., 2021; Luceri et al., 2024; Shafin et al., 2026).

A common approach is to group repeated posts into *duplicate campaigns* and connect accounts that participate in the same campaign. This projection can reveal groups of accounts repeatedly disseminating similar messages. However, duplicate content is not uniformly informative. Some repeated posts contain specific claims, targets, events, or narratives, while others are generic greetings, short praise, vague reactions, platform-generated updates, or reusable expressions. Treating these two forms of repetition as equivalent may create weak edges and distort the interpretation of account-account coordination graphs.

We study the distinction between *generic* and *non-generic* duplicate campaigns. We define semantic specificity as the extent to which a tweet anchors its meaning to identifiable claims, entities, events, policies, accusations, groups, or narratives. A post can be frequently repeated and still be non-generic if it carries a specific political or informational message. Conversely, a post can be fluent or emotionally expressive and still be generic if it can be reused across contexts without changing its meaning. Thus, broad slogans, greetings, motivational quotes, and low-information calls to action are treated as generic unless they are tied to a specific target, event, claim, or narrative frame.

We ask three research questions:

RQ1: How common are generic duplicate campaigns in IO datasets?

RQ2: How do generic and non-generic campaigns differ in size, account participation, textual specificity, and temporal behavior?

RQ3: How does restricting duplicate-based coordination graphs to non-generic campaigns affect graph structure?

This paper makes three contributions. First, we operationalize a generic/non-generic distinction for duplicate campaigns and validate the annotation rubric with independent human verification. Second, we evaluate this distinction on

Proceedings of the 11th Workshop on Natural User-generated Text, pages 210–216
October 28, 2026 ©2026 Association for Computational Linguistics

187,000 English-language tweets from six Russian Twitter Information Operations datasets using three duplicate-detection methods. Third, we show that generic campaigns can substantially inflate the apparent scale of coordination, especially for embedding-based methods, and that filtering to non-generic campaigns yields smaller but denser graphs centered on more narrative-specific repetition.

2 Related Work

Coordination and duplicate-content analysis. Coordinated behavior on social media has been studied through shared behavioral traces, including URLs, hashtags, retweets, temporal synchronization, and content similarity (Pacheco et al., 2021; Nizzoli et al., 2021; Luceri et al., 2024; Minici et al., 2025). Textual repetition provides a particularly accessible signal because it can reveal accounts that disseminate identical or closely related messages using publicly observable content. Prior studies have used exact or near-duplicate text to identify coordinated groups and characterize their dissemination behavior (Pacheco et al., 2020; Vishnuprasad et al., 2024). More recent duplication-focused work has used repeated content to construct campaigns and account-level networks for studying misinformation and information operations (Shafin and Carbunar, 2026; Shafin et al., 2026). These approaches demonstrate the utility of textual similarity for identifying coordinated dissemination, but they primarily determine whether posts are sufficiently similar to be grouped together. They do not explicitly model whether the repeated message is generic and broadly reusable or sufficiently specific to represent a shared claim, event, target, or narrative. Our work addresses this distinction by incorporating semantic specificity into duplicate-campaign analysis.

Noisy user-generated text and semantic similarity. User-generated social-media text poses distinct challenges for NLP because it is often short, informal, non-canonical, and lexically variable. Prior work has shown that normalization choices can affect downstream processing of noisy Twitter text (van der Goot et al., 2017), while evaluations of text representations on Twitter demonstrate that improvements observed on standard text do not necessarily transfer uniformly to noisy user-generated content (Wang et al., 2020). Recent work on sentence embeddings similarly highlights the importance of robustness to lexical variation and non-standard forms in user-generated content (Nishimwe et al., 2024). These challenges are particularly relevant to duplicate detection, where semantically equivalent messages may differ substantially at the surface level. Early work on Twitter paraphrases showed that meaning-preserving messages can be expressed through different lexical forms (Xu et al., 2013), and subsequent work demonstrated the prevalence and importance of lexically divergent paraphrases in short Twitter messages (Xu et al., 2014). The SemEval Paraphrase and Semantic Similarity in Twitter task further formalized both binary paraphrase identification and graded semantic similarity for tweet pairs (Xu et al., 2015). Large-scale paraphrase resources have also been constructed from Twitter by linking independently written tweets that refer to shared URLs (Lan et al., 2017). Collectively, this literature motivates semantic approaches that can recover related messages beyond exact or near-exact lexical overlap. However, semantic similarity alone does not indicate how informative or context-specific the shared content is. Two posts may be highly similar while expressing a reusable greeting, slogan, reaction, or generic call to action rather than a specific narrative.

Semantic specificity and generic repetition. Related NLP research has examined variation in the amount and specificity of information expressed in text. Li and Nenkova (2015) study the identification of content-heavy sentences, distinguishing sentences that convey substantial informational content from those that contribute less content. Li et al. (2016) develop annotation guidelines for sentence specificity and show that specificity can be treated as a systematic linguistic property influenced by both the information expressed in a sentence and its discourse context. Our notion of semantic specificity is related to this literature but serves a different analytical purpose. We operationalize specificity for repeated social-media content by distinguishing generic messages that can be reused across contexts from non-generic messages anchored to identifiable claims, entities, events, policies, accusations, targets, groups, or narratives. This distinction complements prior work on semantic similarity: similarity determines whether messages express related content, whereas specificity characterizes how context-dependent and narratively informative that shared content is. We

Algorithm 1 Generic-Aware Duplicate Campaign Detection

Require: English tweets P, trained tweet classifier M, duplicate methods $\mathcal{D}$
Ensure: Campaign sets and coordination graphs for each method

$P' \leftarrow \emptyset$
for each tweet $p \in P$ **do**
  $t_p \leftarrow \text{NormalizeText}(p.text)$
  if t_p is not empty **then**
    $e_p \leftarrow \text{Embed}(t_p)$
    $\hat{y}_p \leftarrow M(e_p)$
    Add $(p.account, t_p, e_p, \hat{y}_p)$ to P'
  end if
end for
for each duplicate method $d \in \mathcal{D}$ **do**
  $E_d \leftarrow \text{SimilarTweetPairs}(P', d)$
  $C_d \leftarrow \text{ConnectedComponents}(E_d)$
  Remove components with fewer than two accounts from C_d
  for each campaign $c \in C_d$ **do**
    $type(c) \leftarrow$ MajorityLabel of tweet labels in c
  end for
  $G^d_{all} \leftarrow \text{BuildGraph}(C_d)$
  $G^d_{ng} \leftarrow \text{BuildGraph}(\{c \in C_d : type(c) = \text{NON-GENERIC}\})$
end for
return C_d, G^d_{all}, G^d_{ng} for all $d \in \mathcal{D}$

bring these two dimensions together to examine how generic and non-generic repetition is recovered by lexical and embedding-based duplicate-detection methods and how this distinction affects the account-account coordination graphs constructed from duplicate campaigns.

3 Data and Methodology

Data. We use six publicly released Twitter Information Operations datasets attributed to Russian information operations and published by Twitter between 2019 and 2021. We focus on Russian-attributed English-language IO data because these releases provide a public, multi-release benchmark with known platform attribution and enough English text to compare duplicate-detection methods under a controlled setting. We restrict the analysis to English-language tweets, remove empty text, and retain tweet text, account identifiers, timestamps, language labels, hashtags, URLs, and other available metadata. The final corpus contains 187,000 English-language tweet records from 847 accounts. **Tweet labeling and classification pipeline.** We normalize tweet text by removing URLs, mentions, HTML tags, emojis, invisible Unicode characters, and retweet markers; decoding HTML entities; lowercasing; and collapsing whitespace. Hashtag text is retained because it may contain narrative-specific information. We randomly sample 1,000 tweets and label them as GENERIC or NON-GENERIC using an LLM-assisted annotation protocol with GPT-5-mini. The model receives only cleaned tweet text, without account identifiers, dataset names, timestamps, or other metadata. The prompt defines generic tweets as common, reusable, vague, low-information, or not tied to a specific claim, event, target, group, policy, or narrative. It defines non-generic tweets as posts containing a specific informational, political, social, or propagandistic message. We then encode the labeled tweets using sentence-transformers/all-mpnet-base-v2 and train supervised classifiers to scale the labels to the remaining English-language tweets.

Duplicate campaign construction. We define a duplicate campaign as a connected component of tweets from at least two distinct accounts that contain identical or highly similar text. To avoid tuning the duplicate-detection procedure to our downstream graph results, we use three previously proposed duplicate-detection settings and keep their parameters fixed across all datasets.

First, Ratcliff/Obershelp serves as a conservative lexical baseline: following prior coordination studies, tweets are sorted chronologically within each dataset and each tweet is compared with the next ten tweets; pairs from different accounts are linked when similarity is at least 0.70 (Vishnuprasad et al., 2024; Pacheco et al., 2020). This bounded comparison favors temporally proximate lexical repetition and should therefore be interpreted as a conservative baseline rather than an exhaustive all-pairs search. Second, following prior duplicate-campaign work (Shafin and Carbunar, 2026), we encode tweets using paraphrase-multilingual-MiniLM-L12-v2 and apply DBSCAN with min_samples=2 and radius $\epsilon = 1.0$. Third, we use sentence-transformers/all-mpnet-base-v2 embeddings with FAISS cosine similarity (Luceri et al., 2024; Shafin et al., 2026); embeddings are L_2-normalized, indexed with inner-product search, and pairs from different accounts are retained when cosine similarity is at least 0.95.

For all methods, linked tweets are merged into connected components, components with fewer than two accounts are removed, and campaign type is assigned by the majority tweet-level label within each component. In our dataset, no campaign contained equal numbers of generic and non-generic tweets, so every campaign received a unique la-

Table 1: Examples of generic and non-generic duplicate campaign content.

Type	Example Tweet
Generic	"A best friend is someone who came into your life, stayed with you through ups and downs, and never left."
Generic	"No matter how much you feed the wolf, he keeps looking at the forest."
Non-Generic	"The Syrian government and Red Crescent helped return families from al-Hawl camp at Albania's request."
Non-Generic	"If Cuba were racist apartheid-style dictatorship, no one would be so eager to do business with it"

Table 2: Average 5-fold cross-validation performance for generic versus non-generic tweet classification.

Model	Acc.	Generic			Non-Generic			Macro F1	Weighted F1
		P	R	F1	P	R	F1		
Logistic Regression	0.911	0.767	**0.913**	0.833	**0.971**	0.911	0.939	0.886	0.914
Linear SVM	0.908	0.793	0.851	0.818	0.952	0.926	0.938	0.878	0.909
RBF SVM	**0.926**	0.828	0.884	**0.853**	0.962	0.939	**0.951**	**0.902**	**0.927**
Random Forest	0.885	**0.914**	0.581	0.708	0.881	**0.982**	0.928	0.818	0.875
Extra Trees	0.901	0.893	0.672	0.765	0.904	0.974	0.937	0.851	0.896
AdaBoost	0.905	0.827	0.768	0.795	0.928	0.949	0.938	0.867	0.904
Gradient Boosting	0.890	0.840	0.677	0.747	0.903	0.958	0.930	0.839	0.886
KNN	0.903	0.781	0.842	0.808	0.949	0.922	0.935	0.872	0.904
MLP Neural Net	0.920	0.852	0.818	0.831	0.943	0.953	0.948	0.889	0.919

bel. Algorithm 1 summarizes the complete pipeline from tweet-level classification to duplicate campaign construction and downstream graph generation.

Graph construction. We build weighted account-account graphs by connecting two accounts when they participate in the same duplicate campaign; edge weight increases with the number of shared campaigns. Since a campaign with n accounts can induce up to $n(n-1)/2$ pairwise links, large generic campaigns can disproportionately affect graph structure. For each duplicate-detection method, we construct G_{all} using all campaigns and G_{ng} using only non-generic campaigns, and compare nodes, edges, density, average degree, connected components, largest component size, and average edge weight.

4 Findings

4.1 Tweet Classification Results

GPT-5-mini labeled the 1,000 sampled tweets as 241 generic and 759 non-generic. Before training the classifiers, we checked the labeled tweets for exact duplicates after normalization and found that all 1,000 tweets were unique. To assess the reliability of the annotation rubric, two authors independently re-annotated a stratified subset of 200 tweets, consisting of 50 tweets initially labeled as generic and 150 as non-generic by GPT-5-mini. Both annotators used the same generic/non-generic rubric and were blind to the LLM-generated labels and to each other's annotations. They agreed on 194 of the 200 tweets, corresponding to 97.0% raw agreement and a Cohen's κ of 0.919, indicating very high inter-annotator agreement. The annotators agreed on 46 generic and 148 non-generic tweets and disagreed on six tweets. No adjudication was performed before computing inter-annotator agreement. Table 1 presents representative examples of generic and non-generic duplicate content, illustrating the distinction between broadly reusable expressions and context-specific narrative statements.

Table 2 reports the average five-fold performance for the supervised classifiers trained on the 1,000 labeled tweets. The RBF SVM performs best, achieving 0.926 accuracy, 0.902 macro-F1, and 0.927 weighted-F1. We use this model to label the remaining English-language tweets before campaign construction. These scores measure agreement with the generic/non-generic annotation rubric, not correctness against a fully human-labeled gold standard or truthfulness judgments.

4.2 Duplicate Detection Results

The three duplicate-detection methods identify different volumes of duplicate campaigns. Ratcliff/Obershelp detects 2,403 duplicate campaigns, making it the most conservative method. Embedding-based clustering detects 7,651 duplicate campaigns, while embedding-based cosine similarity detects 7,527 duplicate campaigns. Thus,

Table 3: Campaign-level characteristics across duplicate-detection methods. Duration is measured in days. Hashtag and URL rates are the percentage of campaigns containing at least one tweet with a hashtag or URL.

Method	Type	Campaigns	Tweets/ Campaign	Accounts/ Campaign	Max Accounts	Unique Accounts	Words/ Text	Avg. Duration	Hashtag %	URL %
Ratcliff/Obershelp	Generic	109	2.17	2.13	4	116	5.91	0.12	7.34	33.94
	Non-Generic	2,294	2.80	2.65	33	268	11.09	0.03	11.46	5.27
Embedding + Clustering	Generic	2,961	4.28	2.71	21	373	8.59	280.29	4.19	28.54
	Non-Generic	4,690	5.06	4.80	46	348	10.85	17.22	8.91	6.12
Embedding + Similarity	Generic	2,949	4.29	2.71	20	375	8.67	273.94	4.27	28.86
	Non-Generic	4,578	5.35	4.89	76	362	11.04	18.01	9.79	6.62

Table 4: Account-account coordination graph characteristics across duplicate-detection methods and campaign types.

Method	Campaign Type	Nodes	Edges	Density	Avg. Degree	Components	Largest Comp.	Avg. Edge Weight
Ratcliff/Obershelp	Both Generic and Non-Generic	312	4,086	0.084	26.19	40	138	2.14
	Only Non-Generic	266	4,027	0.114	30.28	30	138	2.14
Embedding + Clustering	Both Generic and Non-Generic	497	12,020	0.098	48.37	16	428	8.93
	Only Non-Generic	348	10,433	0.173	59.96	25	166	9.39
Embedding + Cosine Similarity	Both Generic and Non-Generic	503	12,096	0.096	48.10	16	435	9.62
	Only Non-Generic	362	10,528	0.161	58.17	26	207	10.17

the embedding-based methods recover more than three times as many duplicate campaigns as the lexical baseline.

The generic share also differs sharply across methods. Ratcliff/Obershelp identifies 109 generic campaigns and 2,294 non-generic campaigns, so generic campaigns account for only 4.5% of its detected campaigns. In contrast, embedding-based clustering identifies 2,961 generic and 4,690 non-generic campaigns, while embedding-based cosine similarity identifies 2,949 generic and 4,578 non-generic campaigns. Generic campaigns therefore account for 38.7% and 39.2% of campaigns detected by the two embedding-based methods, respectively. This suggests that semantic duplicate detection is more likely to surface reusable, low-information repetition in addition to narrative-specific duplication.

4.3 Campaign-Level Differences

Table 3 compares generic and non-generic campaigns across three detection methods. Across all three methods, non-generic campaigns contain more tweets per campaign and involve more accounts per campaign. Under embedding-based similarity, for example, non-generic campaigns average 5.35 tweets and 4.89 accounts per campaign, compared with 4.29 tweets and 2.71 accounts for generic campaigns.

The textual and temporal characteristics reinforce this distinction. Non-generic campaigns have longer representative text across all methods, suggesting greater narrative specificity. Under embedding-based similarity, non-generic campaigns average 11.04 words per representative text, compared with 8.67 for generic campaigns. Generic campaigns, however, have longer average duration and higher URL prevalence under the embedding-based methods. This suggests that reusable low-information content can recur across broader time spans and may be more strongly associated with URL-sharing behavior, whereas non-generic campaigns are more textually specific and more concentrated in time.

4.4 Effect on Coordination Graphs

Table 4 compares account-account coordination graphs constructed from all campaigns with graphs constructed only from non-generic campaigns. For Ratcliff/Obershelp, restricting the graph to non-generic campaigns removes 46 nodes but only 59 edges, and the largest connected component remains unchanged at 138 nodes. This suggests that generic Ratcliff/Obershelp campaigns mostly involve peripheral accounts rather than the main coordination core.

The embedding-based methods show a much stronger effect. With embedding clustering, restricting the graph to non-generic campaigns reduces the largest connected component from 428 to 166 nodes, while density increases from 0.098 to 0.173 and average degree increases from 48.37 to 59.96. The FAISS similarity method shows the same pattern: the largest component decreases from 435 to 207 nodes, while density rises from 0.096 to 0.161. These results indicate that generic dupli-

cate campaigns can increase the apparent scale of duplicate-based coordination networks, especially by connecting many accounts into large components. After filtering to non-generic campaigns, the resulting graphs are smaller and denser, consistent with a more concentrated network around narrative-specific repetition. Because graph density can also increase when peripheral nodes are removed, we interpret this structural change as evidence of concentration rather than, by itself, stronger coordination.

5 Discussion and Limitations

Our results show that duplicate content should not be treated as uniformly informative evidence of coordination. The effect varies by method. Ratcliff/Obershelp identifies relatively few generic campaigns, while embedding-based methods recover many generic or reusable posts. This broader recall is useful, but it also increases the risk of treating low-information repetition as meaningful narrative coordination.

The long duration of generic campaigns under the embedding-based methods is particularly important. Generic text can be semantically similar across long time spans, causing embedding-based methods to connect posts that are reusable but not necessarily part of the same narrative push. This does not mean such campaigns are irrelevant; generic content may support account maintenance, engagement, or visibility. However, it should be separated from non-generic duplicate campaigns when interpreting coordination graphs.

This study has limitations. First, we analyze only English-language Russian IO datasets. The 39% generic-campaign rate should therefore not be interpreted as universal; it may vary across languages, platforms, actors, and campaign periods. Our claim is not that this corpus represents all IO behavior, but that generic repetition can materially affect duplicate-based coordination graphs even in a widely used public IO setting. Second, the classifier is trained on LLM-assisted rubric labels rather than a fully human-labeled gold standard. Although two authors independently re-annotated a stratified subset of 200 tweets and achieved 97.0% raw agreement (Cohen's $\kappa = 0.919$), systematic biases in the LLM-assisted labels could still propagate to the remaining tweets and downstream graph analysis. Third, although the labeled tweets were unique after normalization, future work should evaluate campaign-level or account-level splits to test generalization across near-duplicate groups and accounts. Finally, because this short paper uses duplicate-detection parameters from prior work, we do not conduct a full threshold sensitivity analysis; future work should examine threshold robustness, per-release graph statistics, temporal burstiness, recurrence, and synchronization.

6 Conclusion

This paper studies generic duplicate campaigns in duplicate-based coordination analysis of information operations. Using 187,000 English-language tweets from six Russian Twitter Information Operations datasets, we classify tweets as generic or non-generic, construct duplicate campaigns using three methods, and compare coordination graphs built from all campaigns with graphs built only from non-generic campaigns. Generic campaigns account for nearly 39% of embedding-based duplicate campaigns and substantially affect graph structure. Restricting graphs to non-generic campaigns produces smaller but denser graphs with smaller largest connected components, suggesting that filtering generic campaigns provides a more concentrated view of narrative-specific repetition. These findings highlight a simple but important refinement: repeated content should be evaluated not only by similarity, but also by semantic specificity.

References

Wuwei Lan, Siyu Qiu, Hua He, and Wei Xu. 2017. A continuously growing dataset of sentential paraphrases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1224–1234, Copenhagen, Denmark. Association for Computational Linguistics.

Junyi Jessy Li and Ani Nenkova. 2015. Detecting content-heavy sentences: A cross-language case study. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1271–1281, Lisbon, Portugal. Association for Computational Linguistics.

Junyi Jessy Li, Bridget O'Daniel, Yi Wu, Wenli Zhao, and Ani Nenkova. 2016. Improving the annotation of sentence specificity. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3921–3927, Portorož, Slovenia. European Language Resources Association (ELRA).

Luca Luceri, Valeria Pant'e, Keith Burghardt, and Emilio Ferrara. 2024. Unmasking the web of de-

ceit: Uncovering coordinated activity to expose information operations on twitter. In *Proceedings of the ACM Web Conference 2024*, pages 2530–2541. Association for Computing Machinery.

Marco Minici, Luca Luceri, Francesco Fabbri, and Emilio Ferrara. 2025. IOHunter: Graph foundation model to uncover online information operations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27):28258–28266.

Lydia Nishimwe, Benoît Sagot, and Rachel Bawden. 2024. Making sentence embeddings robust to user-generated content. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10984–10998, Torino, Italia. ELRA and ICCL.

Leonardo Nizzoli, Serena Tardelli, Marco Avvenuti, Stefano Cresci, and Maurizio Tesconi. 2021. Coordinated behavior on social media in 2019 uk general election. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):443–454.

Diogo Pacheco, Alessandro Flammini, and Filippo Menczer. 2020. Unveiling coordinated groups behind white helmets disinformation. In *Companion Proceedings of the Web Conference 2020*, pages 611–616. Association for Computing Machinery.

Diogo Pacheco, Pik-Mai Hui, Christopher Torres-Lugo, Bao Tran Truong, Alessandro Flammini, and Filippo Menczer. 2021. Uncovering coordinated networks on social media: Methods and case studies. *Proceedings of the International AAAI Conference on Web and Social Media*, 15(1):455–466.

Ruben Recabarren, Bogdan Carbunar, Nestor Hernandez, and Ashfaq Ali Shafin. 2023. Strategies and vulnerabilities of participants in venezuelan influence operations. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 6683–6700.

Ashfaq Ali Shafin and Khandaker Mamun Ahmed. 2025a. The language of influence: Sentiment, emotion, and hate speech in state sponsored influence operations. In *Proceedings of the 18th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '25, page 506–511.

Ashfaq Ali Shafin and Khandaker Mamun Ahmed. 2025b. Toxicity in state sponsored information operations. In *Proceedings of the 36th ACM Conference on Hypertext and Social Media*, HT '25, page 86–90.

Ashfaq Ali Shafin and Bogdan Carbunar. 2026. Duplicating deceit: Inauthentic behavior among indian misinformation duplicators on x/twitter. In *Social Networks Analysis and Mining: 17th International Conference, ASONAM 2025, Proceedings, Part II*, volume 16323 of *Lecture Notes in Computer Science*, pages 363–371. Springer.

Ashfaq Ali Shafin, Md Nahid Siddique, and Bogdan Carbunar. 2026. IO-X: Detecting and attributing content-duplicating influence operations on X (Twitter). In *Proceedings of the ACM Web Conference 2026*, pages 1809–1818.

Rob van der Goot, Barbara Plank, and Malvina Nissim. 2017. To normalize, or not to normalize: The impact of normalization on part-of-speech tagging. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 31–39, Copenhagen, Denmark. Association for Computational Linguistics.

Padinjaredath Suresh Vishnuprasad, Gianluca Nogara, Felipe Cardoso, Stefano Cresci, Silvia Giordano, and Luca Luceri. 2024. Tracking fringe and coordinated activity on twitter leading up to the us capitol attack. *Proceedings of the International AAAI Conference on Web and Social Media*.

Lili Wang, Chongyang Gao, Jason Wei, Weicheng Ma, Ruibo Liu, and Soroush Vosoughi. 2020. An empirical survey of unsupervised text representation methods on Twitter data. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020)*, pages 209–214, Online. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Bill Dolan. 2015. SemEval-2015 task 1: Paraphrase and semantic similarity in Twitter (PIT). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 1–11, Denver, Colorado. Association for Computational Linguistics.

Wei Xu, Alan Ritter, Chris Callison-Burch, William B. Dolan, and Yangfeng Ji. 2014. Extracting lexically divergent paraphrases from Twitter. *Transactions of the Association for Computational Linguistics*, 2:435–448.

Wei Xu, Alan Ritter, and Ralph Grishman. 2013. Gathering and generating paraphrases from Twitter with application to normalization. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 121–128, Sofia, Bulgaria. Association for Computational Linguistics.

Savvas Zannettou, Tristan Caulfield, Emiliano De Cristofaro, Michael Sirivianos, Gianluca Stringhini, and Jeremy Blackburn. 2019. Disinformation warfare: Understanding state-sponsored trolls on twitter and their influence on the web. In *Companion Proceedings of The 2019 World Wide Web Conference*, WWW '19, page 218–226, New York, NY, USA. Association for Computing Machinery.

Personalized Emotion Analysis via Prompting Large Language Models with Writer Information

Kentaro Hanafusa[1] **Daisuke Maekawa**[1] **Tomoyuki Kajiwara**[1,2]
[1]Ehime University [2]The University of Osaka
{hanafusa@ai., maekawa@ai., kajiwara@}cs.ehime-u.ac.jp

Abstract

Personalized emotion analysis aims to account for individual differences in emotional expression by adapting predictions to each writer. Existing personalization methods typically require dedicated feature extraction modules or modifications to model architectures, increasing model complexity. In this paper, we propose a prompt-based personalization framework for emotion analysis that incorporates writer information into the input prompt of a large language model (LLM), without requiring fine-tuning or architectural changes. Specifically, we investigate two complementary types of writer information: personality traits representing stable writer characteristics and post history representing writer-specific patterns of emotional expression. Experimental results on the WRIME corpus in Japanese show that incorporating temporally close post history consistently improves emotion intensity estimation across multiple LLMs. Personality information alone provides limited improvements, but yields additional gains for several models when combined with post history. These findings suggest that prompt-based personalization is a practical alternative to architecture-level personalization for LLM-based emotion analysis.

1 Introduction

Emotion analysis of user-generated text remains a challenging task because emotional expressions are inherently subjective and vary across individuals (Kajiwara et al., 2021; Milkowski et al., 2021). The same expression may convey different emotional intensities depending on the writer's personality, writing style, or habitual patterns of emotional expression (Suzuki et al., 2022b; Lyu et al., 2023). As large language models (LLMs) are increasingly used to analyze user-generated content, modeling such user-specific variation has become an important challenge. Consequently, personalized emotion analysis, which adapts predictions to individual writers, has emerged as a promising direction for improving LLM-based emotion analysis.

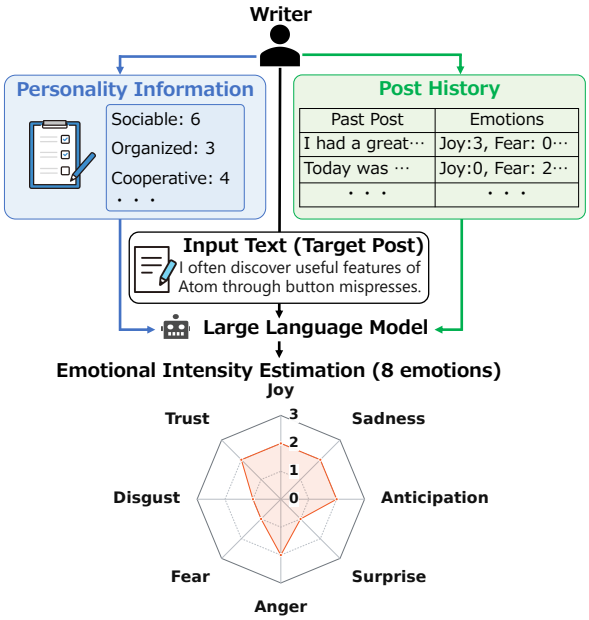


Figure 1: Overview of the proposed method.

Previous studies have incorporated writer-specific information, such as personality traits or representations derived from historical posts, into neural emotion analysis models (Suzuki et al., 2022b; Lyu et al., 2023). While these approaches have demonstrated the usefulness of writer information, they typically rely on dedicated user encoders, feature extraction modules, or attention mechanisms beyond the underlying language model (Chen et al., 2016; Wu et al., 2018; Milkowski et al., 2021; Zhang et al., 2024). Such model-specific designs increase architectural complexity and reduce portability across different LLMs.

Recent advances in LLM prompting provide a simple alternative. Instead of modifying model architectures, writer-specific information can be incorporated directly into prompts as natural language context. This naturally raises the following question: Can prompting alone effectively personalize

Proceedings of the 11th Workshop on Natural User-generated Text, pages 217–226
October 28, 2026 ©2026 Association for Computational Linguistics

LLMs for emotion analysis on user-generated text?

In this paper, as shown in Figure 1, we propose a prompt-based personalization framework that injects writer information into LLM prompts without architectural modifications or fine-tuning. We consider two complementary sources of writer information: personality, which represents relatively stable user characteristics, and post history, which captures user-specific patterns of emotional expression and their temporal context. Because both are provided entirely through prompts, the proposed approach can be readily applied across LLMs without introducing dedicated personalization components.

We evaluate the proposed method on WRIME (Kajiwara et al., 2021; Suzuki et al., 2022a), a benchmark corpus of Japanese social media posts annotated with writers' subjective emotional intensities. Experimental results demonstrate that temporally close post history consistently improves emotion estimation across multiple LLMs, while personality information provides complementary gains for several models. These findings suggest that prompting with writer information is an effective, lightweight approach to LLM personalization for user-generated text.

Our contributions are summarized as follows.

- We propose a prompt-based personalization framework that incorporates writer information into LLMs for emotion analysis without requiring model modification or fine-tuning.
- We systematically investigate two complementary forms of writer information—personality and post history—and analyze their respective roles in personalized emotion analysis.
- We demonstrate across multiple LLMs that temporally close post history consistently improves emotion estimation, while personality information provides complementary benefits when combined with post history.

2 Related Work

Early studies on personalized emotion analysis focused on movie and product reviews and proposed methods that combine textual feature representations with representations of writers and products (Tang et al., 2015; Chen et al., 2016; Wu et al., 2018). These studies personalize sentiment polarity classification by learning embedding representations for individual writers and products and integrating them with textual representations through concatenation or attention mechanisms.

Personalized emotion analysis using pretrained language models has also been investigated. Milkowski et al. (2021) combined Personal Emotional Bias (PEB), which represents individual biases in emotion annotations, with textual representations obtained from BERT to perform emotion estimation while accounting for individual differences. In addition, Lyu et al. (2023) proposed a method that constructs user and product representations from past reviews associated with the same user or product and models their relationships using an attention mechanism. By explicitly using past reviews, their method improved sentiment polarity classification performance across multiple pretrained language models.

For personalized emotion analysis using writers' personality information, Suzuki et al. (2022b) proposed an emotional intensity estimation method that integrates personality assessment results based on the Big Five (Goldberg, 1992) with textual representations obtained from BERT, using the WRIME corpus of Japanese SNS posts. Their experimental results demonstrated that incorporating personality information improves the performance of estimating writers' subjective emotional intensities.

Recently, methods have also been proposed to efficiently incorporate personalized information into pretrained language models. Zhang et al. (2024) proposed E2LN, which injects user and product information into Layer Normalization and enables both the modeling of long review texts and the incorporation of personalized information through parameter-efficient fine-tuning. Although this method reduces the number of trainable parameters compared with conventional personalization approaches, it still requires introducing a dedicated mechanism for processing personalized information and training the model.

These existing studies require dedicated mechanisms for encoding writer-specific information and integrating it with textual representations, as well as fine-tuning on emotion analysis data. In contrast, we propose a personalization method that directly incorporates personality information and post history into the LLM prompt, eliminating the need for specialized feature extraction mechanisms, modifications to the model architecture, or fine-tuning.

3 Methods

We perform personalized emotion analysis by incorporating writer information into the prompt of

an LLM. Specifically, we consider two complementary forms of writer information: personality, representing stable characteristics of the writer, and post history, capturing writer-specific patterns of emotional expression. The input therefore consists of the target post together with either personality information or post history, or both.

3.1 Personality-based Personalization

The writer's personality is expected to influence not only how emotions are experienced but also how they are expressed in text. Therefore, we augment the base prompt with personality information corresponding to the writer of the target post and instruct the LLM to estimate emotional intensity while taking the writer's personality into account.

In this study, we use the Big Five personality model (Goldberg, 1992) to represent personality information. The Big Five model characterizes an individual's personality along five traits: extraversion, agreeableness, conscientiousness, neuroticism, and openness. This personality information is obtained from a 60-item personality assessment based on the Big Five model (Saito et al., 2001) and is provided for each writer in WRIME. Each item consists of a personality-descriptive word, such as "talkative" or "anxious," and writers rate how well each word describes them on a seven-point scale. Continuous scores indicating the degree to which each of the five traits applies to the writer are then calculated from these responses.

To investigate the effect of the level of detail in personality information, we use the following two input formats.

- **Personality60**: The response values for the 60 personality assessment items are provided in a format such as "talkative: 6, anxious: 2." This format provides the LLM with detailed personality information before it is aggregated into the five traits.
- **Big Five**: The five traits—extraversion, agreeableness, conscientiousness, neuroticism, and openness—and their corresponding scores are provided as input. This format presents the writer's personality to the LLM as a summary along five major dimensions.

3.2 Post History-based Personalization

The relationship between specific expressions and emotional intensities is expected to vary across writers. Therefore, we include multiple posts written by the same writer and their emotion labels in the prompt as few-shot in-context learning examples, enabling the LLM to capture writer-specific patterns of emotional expression.

We use the following two post-history selection methods.

- **Random**: Ten posts are randomly selected from those written by the same writer.
- **Recent**: Among the posts written by the same writer before the target post, the ten temporally closest posts are selected.

Recent uses only posts available at prediction time, reflecting the writer's recent emotions and topics. This prevents future information leakage and reflects practical prediction scenarios.

3.3 Combining Personality and Post History

Personality information represents stable characteristics of the writer, whereas post history reflects specific patterns of emotional expression. We therefore incorporate both types of information into the prompt and use them in a complementary manner.

We evaluate **Personality60 + Recent**, which combines the 60-item personality information with post history, and **Big Five + Recent**, which combines the five Big Five traits with post history.

4 Experiments

4.1 Settings

Dataset We evaluate our method using WRIME[1] (Kajiwara et al., 2021; Suzuki et al., 2022a), a Japanese emotion analysis corpus. WRIME contains Japanese SNS posts annotated with writers' subjective emotions and readers' objective emotions. We focus on the writers' subjective emotions and estimate the intensities of Plutchik's eight basic emotions (Plutchik, 1980): joy, sadness, anticipation, surprise, anger, fear, disgust, and trust. Emotion intensity is represented on a four-point scale, $\{0, 1, 2, 3\}$, corresponding to no, weak, medium, and strong, respectively.

For the data split, 7,000 posts from WRIME are used to construct writer-specific information, including post history, and to train the personality feature extractor used in the comparison method (Suzuki et al., 2022b). The remaining posts are divided into 24,000 training instances, 2,000 validation instances, and 2,000 test instances. The

[1] https://github.com/ids-cv/wrime

Split	Writers	Posts
Feature extraction	60	7,000
Training	40	24,000
Validation	10	2,000
Test	10	2,000

Table 1: Data splits of the WRIME corpus.

training, validation, and test sets contain posts from 40, 10, and 10 writers, respectively, with no overlap in writers across the splits. Table 1 presents the dataset statistics.

Models As a baseline, we use BERT[2] (Devlin et al., 2019), which was pretrained on Japanese Wikipedia. For the evaluated LLMs, we use LLM-jp[3] (LLM-jp, 2024), which was pretrained on Japanese and English data as well as source code, and Swallow[4][5] (Fujii et al., 2024; Ma et al., 2025), which was obtained by continual pretraining of an English LLM on Japanese data. We also use Qwen[6] (Qwen Team, 2026) and Gemma[7] (Gemma Team, 2026), multilingual models pretrained on text in multiple languages, primarily English.

Comparison Methods We use Japanese BERT as the baseline. For comparison, we employ a basic model that estimates emotional intensity solely from the input post and the personalized method (Personality) proposed by Suzuki et al. (2022b).

In the baseline implementation, the final-layer representation corresponding to the initial special token `[CLS]` is used as the feature vector for the post, and a fully connected layer predicts one of four emotional intensity labels. We construct an independent classifier for each of the eight basic emotions and fine-tune the pretrained model on the WRIME training data. We set the learning rate to 5.0×10^{-6}, the dropout rate to 0.1, and the batch size to 32, and use AdamW (Loshchilov and Hutter, 2019) for optimization. We also apply early stopping and terminate training when performance on the validation data does not improve for three consecutive epochs.

In the personalized comparison method (Personality), the responses to the 60-item personality assessment based on the Big Five personality model are transformed into a personality representation. The resulting personality representation is integrated with the textual representation obtained from BERT to estimate emotional intensity while taking the writer's personality into account.

In the experiments using LLMs, we compare the following settings.

- **0-shot**: Only the instruction and the target post are provided, without writer-specific information or examples with gold labels.
- **10-shot**: Ten posts and their gold emotion labels, randomly selected from the training data without distinguishing writers, are provided as few-shot in-context learning examples.
- **Personality60**: The writer's responses to the 60-item personality assessment are added to the prompt. Because no examples with gold labels are included, this setting corresponds to a 0-shot setting.
- **Big Five**: The five traits—extraversion, agreeableness, conscientiousness, neuroticism, and openness—and their corresponding scores are added to the prompt. This is also a 0-shot setting because it does not include examples with gold labels.
- **Random**: Ten posts are randomly selected from those written by the same writer as the target post, and the post texts and their gold emotion labels are provided as few-shot in-context learning examples.
- **Recent**: Among the posts written by the same writer before the target post, the ten posts whose timestamps are closest to that of the target post are provided as few-shot in-context learning examples.
- **Personality60 + Recent**: The 60-item personality information and the post history selected by Recent are provided simultaneously.
- **Big Five + Recent**: The five Big Five traits and the post history selected by Recent are provided simultaneously.

[2] https://huggingface.co/tohoku-nlp/bert-base-japanese

[3] https://huggingface.co/llm-jp/llm-jp-4-8b-thinking

[4] https://huggingface.co/tokyotech-llm/Qwen3-Swallow-8B-SFT-v0.2

[5] https://huggingface.co/tokyotech-llm/Qwen3-Swallow-32B-SFT-v0.2

[6] https://huggingface.co/Qwen/Qwen3.6-27B

[7] https://huggingface.co/google/gemma-4-31B-it

Model	Method	Joy	Sadness	Anticipation	Surprise	Anger	Fear	Disgust	Trust	Overall
BERT	Fine-tuning	**0.524**	0.417	0.298	0.286	**0.399**	0.282	0.412	0.143	0.410
	+ Personality	0.518	**0.438**	**0.315**	**0.326**	0.365	**0.294**	**0.424**	**0.225**	**0.424**
llm-jp-8B	0-shot	**0.564**	0.499	0.362	0.339	**0.403**	**0.429**	0.354	0.209	**0.456**
	+ Big Five	0.546	**0.507**	**0.381**	**0.340**	0.357	0.351	**0.379**	**0.214**	0.448
	+ Personality60	0.481	0.423	0.341	0.295	0.266	0.265	0.327	0.181	0.376
Swallow-8B	0-shot	**0.565**	0.453	0.355	0.298	**0.350**	0.353	0.201	**0.221**	0.431
	+ Big Five	0.553	**0.464**	**0.423**	**0.318**	0.304	**0.368**	**0.309**	0.199	**0.435**
	+ Personality60	0.521	0.438	0.390	0.306	0.317	0.343	0.271	0.164	0.416
Swallow-32B	0-shot	**0.599**	**0.507**	0.385	0.328	**0.405**	0.382	0.343	0.165	**0.464**
	+ Big Five	0.569	0.495	**0.426**	0.323	0.331	**0.394**	**0.375**	0.162	0.454
	+ Personality60	0.540	0.493	0.393	**0.356**	0.354	0.347	0.348	**0.210**	0.448
Gemma-31B	0-shot	**0.676**	0.536	**0.498**	**0.494**	**0.548**	0.417	**0.477**	**0.295**	**0.543**
	+ Big Five	0.671	0.540	0.496	0.484	0.525	**0.458**	0.440	0.269	0.531
	+ Personality60	0.656	**0.564**	0.470	0.462	0.510	**0.458**	0.433	0.223	0.513
Qwen-27B	0-shot	**0.631**	0.468	0.417	0.441	0.519	0.398	0.384	0.180	0.505
	+ Big Five	**0.631**	**0.514**	0.427	0.435	**0.530**	**0.418**	**0.446**	**0.238**	**0.518**
	+ Personality60	0.611	0.512	**0.441**	**0.459**	0.415	0.339	0.411	0.208	0.483

Table 2: Results of writer emotion intensity estimation with personality information. Personality denotes an existing method that incorporates personality information (Suzuki et al., 2022b). Overall denotes the QWK computed over all eight emotions jointly. Bold values indicate the best performance for each model.

The 0-shot and 10-shot settings serve as LLM baselines that do not use writer-specific information. By comparing 10-shot with Random, we examine the effect of selecting examples written by the same writer. By comparing Random with Recent, we further examine the effect of selecting post history while considering temporal proximity.

Metric We use Quadratic Weighted Kappa (QWK) (Cohen, 1968), a standard evaluation metric for ordinal classification tasks.

4.2 Results

4.2.1 Personality-based Personalization

Table 2 presents the evaluation results for personalized methods using personality information. We compare the existing method based on Japanese BERT with three settings for each LLM: the 0-shot setting, which uses only the input post; Big Five, which provides the five Big Five traits; and Personality60, which provides the responses to the 60-item personality assessment.

For BERT, incorporating personality information improved the overall QWK across the eight emotions, confirming the effectiveness of personality information in conventional training-based methods. For the LLMs, however, the effect of personality information varied across models. Big Five improved the overall QWK across the eight emotions for Swallow-8B and Qwen-27B, whereas the 0-shot setting achieved the best performance for the other models. These results suggest that incorporating personality information through prompting is effective for some LLMs, but its effectiveness depends on the characteristics of the model.

A comparison between Big Five and Personality60 shows that Big Five achieved a higher overall QWK across the eight emotions for every LLM. Moreover, Personality60 performed worse than the 0-shot setting for all LLMs. These results suggest that presenting personality information summarized into five major traits may allow LLMs to utilize information relevant to emotion estimation more effectively than directly providing all 60 detailed items. The performance degradation observed with Personality60 may also be attributable to redundant input information and the increased prompt length.

At the emotion level, incorporating the Big Five traits outperformed the 0-shot setting for many models in sadness, anticipation, fear, and disgust. Relatively large improvements were observed for disgust with Swallow-8B and for sadness, disgust, and trust with Qwen-27B. In contrast, no model showed an improvement for joy, and only a limited number of models improved for anger. These findings indicate that the effectiveness of personality information varies across emotions: it is relatively effective for estimating sadness, anticipation, fear, and disgust, but is not necessarily effective for joy or anger.

Model	Method	Joy	Sadness	Anticipation	Surprise	Anger	Fear	Disgust	Trust	Overall
BERT	Fine-tuning	**0.524**	0.417	0.298	0.286	**0.399**	0.282	0.412	0.143	0.410
	+ Personality	0.518	**0.438**	**0.315**	**0.326**	0.365	**0.294**	**0.424**	**0.225**	**0.424**
llm-jp-8B	0-shot	0.564	0.499	0.362	0.339	0.403	0.429	0.354	**0.209**	0.456
	10-shot	0.587	0.530	0.359	0.347	**0.467**	0.419	0.352	0.199	0.479
	Random	0.601	**0.544**	0.369	0.399	0.450	**0.473**	0.362	0.178	0.489
	Recent	**0.618**	0.532	0.374	**0.412**	0.463	0.462	0.380	0.167	**0.496**
	+ Personality60	0.594	0.540	0.374	0.393	0.447	0.449	**0.391**	0.176	0.485
	+ Big Five	0.605	0.539	**0.391**	0.392	0.462	0.469	0.386	0.201	**0.496**
Swallow-8B	0-shot	0.565	0.453	**0.355**	**0.298**	0.350	0.353	0.201	**0.221**	0.431
	10-shot	0.532	0.495	0.291	0.269	0.479	0.365	0.200	0.083	0.413
	Random	0.530	**0.523**	0.334	0.274	0.433	0.427	0.282	0.078	0.422
	Recent	0.598	0.513	0.328	0.363	0.506	0.477	0.405	0.101	0.471
	+ Personality60	**0.588**	**0.511**	0.321	0.329	**0.462**	0.477	**0.379**	0.116	**0.459**
	+ Big Five	0.594	0.504	0.307	0.351	0.498	**0.500**	0.393	0.094	0.465
Swallow-32B	0-shot	0.599	0.507	0.385	0.328	0.405	0.382	0.343	**0.165**	0.464
	10-shot	0.603	0.535	0.381	0.340	0.428	0.430	0.289	0.164	0.476
	Random	0.619	0.537	**0.408**	0.366	0.429	0.441	0.209	0.143	0.478
	Recent	**0.635**	0.519	0.350	**0.409**	**0.467**	0.456	0.359	0.132	**0.490**
	+ Personality60	0.586	**0.544**	0.333	0.392	0.448	0.417	0.309	0.118	0.465
	+ Big Five	0.614	0.537	0.340	**0.419**	0.438	0.443	0.314	0.138	0.479
Gemma-31B	0-shot	0.676	0.536	**0.498**	0.494	0.548	0.417	0.477	0.295	0.543
	10-shot	0.667	0.529	0.431	0.497	**0.603**	0.410	0.494	0.247	0.530
	Random	0.671	0.534	0.461	0.489	0.572	0.446	0.525	0.206	0.535
	Recent	0.695	0.540	0.462	**0.523**	0.584	0.443	0.559	0.304	0.564
	+ Personality60	0.696	**0.548**	0.491	0.517	0.586	0.460	0.560	0.300	0.570
	+ Big Five	**0.697**	0.540	0.497	0.510	0.599	**0.471**	**0.569**	**0.311**	**0.572**
Qwen-27B	0-shot	0.631	0.468	0.417	0.441	0.519	0.398	0.384	**0.180**	0.505
	10-shot	0.639	0.493	0.399	0.440	0.512	0.403	0.373	0.115	0.498
	Random	0.647	0.480	0.410	0.415	0.511	0.430	0.373	0.122	0.496
	Recent	**0.672**	0.495	0.414	0.452	**0.574**	0.423	0.411	0.095	0.513
	+ Personality60	0.665	**0.536**	**0.441**	**0.473**	0.537	**0.460**	**0.448**	0.159	**0.526**
	+ Big Five	0.667	0.520	0.419	0.455	0.558	0.452	0.432	0.151	0.521
Annotator	A	0.490	0.314	0.377	0.339	0.255	0.354	0.305	**0.135**	0.367
	B	**0.590**	0.436	**0.438**	**0.400**	0.285	0.363	**0.448**	0.118	**0.449**
	C	0.553	**0.485**	0.426	0.347	**0.302**	**0.365**	0.275	0.131	0.426

Table 3: Results of writer emotion intensity estimation using writer information in terms of QWK. Personality denotes an existing method that incorporates personality information (Suzuki et al., 2022b). Overall denotes the QWK computed over all eight emotions jointly. Bold values indicate the best performance for each model.

4.2.2 Post History-based Personalization

Table 3 presents the results of methods using post history, methods combining post history with personality information, and human evaluation. The annotator scores represent the agreement between each reader's objective emotion annotations and the writer's subjective emotion labels in terms of QWK, without access to the writer's personality information or post history. For each LLM, we compare 0-shot, 10-shot with examples randomly selected from the training data, Random with examples randomly selected from posts written by the same writer, and Recent with temporally close posts written before the input post. We further combine Recent with either Personality60 or Big Five to investigate the complementary use of post history and personality information.

Random outperformed the standard 10-shot setting for many models. This result indicates that using posts written by the same writer as few-shot examples enables LLMs to capture writer-specific emotional expressions and writing styles. Furthermore, Recent outperformed 0-shot for all LLMs and achieved the highest overall QWK across the eight emotions for several models. This suggests that, even among examples written by the same writer, selecting temporally close posts written before the input post effectively reflects the writer's recent emotional expressions and context. The comparison with the standard 10-shot setting also confirms that, in addition to the number of examples, the correspondence between the examples and the writer as well as their temporal proximity are important in few-shot learning.

At the emotion level, Recent improved performance for a wide range of emotions, including joy,

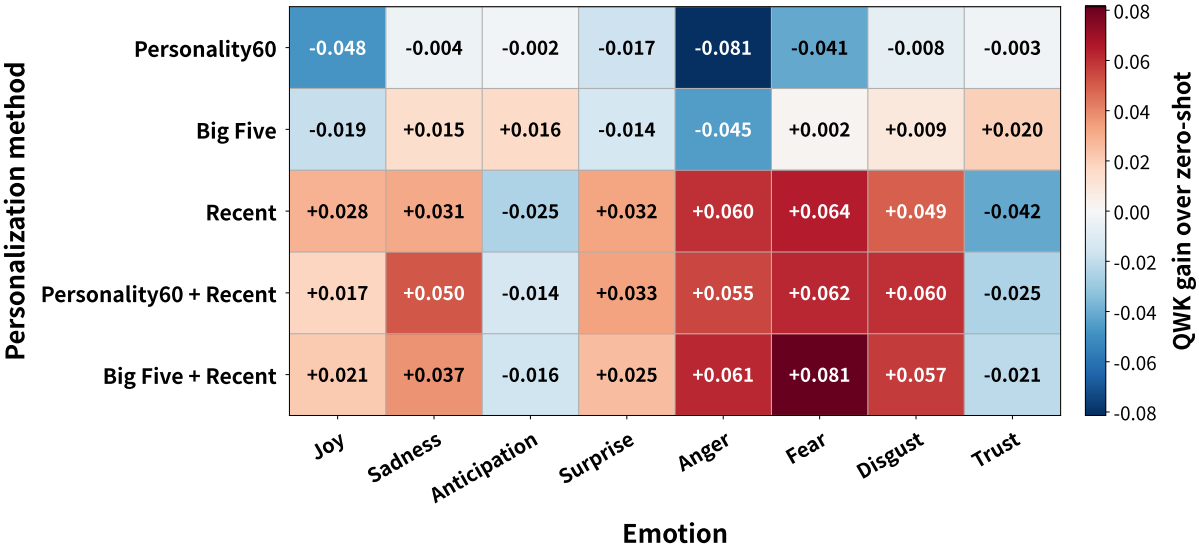


Figure 2: Emotion-wise QWK improvements achieved by each personalization method, averaged across the evaluated LLMs. Improvements are calculated relative to the 0-shot setting for each model.

sadness, surprise, anger, fear, and disgust. In particular, compared with Random, Recent achieved higher performance for all models on joy, surprise, and disgust. These results suggest that temporally close post history is useful not only for capturing writer-specific emotional expressions but also for distinguishing emotions relevant to the target post.

When post history was combined with personality information, the overall QWK across the eight emotions exceeded that of Recent alone for Gemma-31B and Qwen-27B. In particular, the combination with Big Five performed well for Gemma-31B, whereas the combination with Personality60 performed well for Qwen-27B. This may be because post history captures the writer's recent emotional expressions and specific expressive tendencies, while personality information complements them by representing the writer's relatively stable characteristics. Thus, combining post history with personality information enables the model to capture writer-specific characteristics from different perspectives and yields additional improvements that cannot be obtained from post history alone.

Compared with human evaluation, the best overall QWK across the eight emotions achieved by each LLM exceeded the highest score among the human annotators. This result demonstrates that the proposed method can achieve high estimation performance for writers' subjective emotional intensities, which are difficult for readers to infer, by leveraging post history and personality information as additional cues. Overall, incorporating temporally close post history from the same writer into the prompt consistently improves performance across multiple LLMs. Moreover, combining post history with personality information enables the model to capture the writer's stable characteristics and temporally varying patterns of emotional expression in a complementary manner.

5 Analysis

5.1 Effectiveness across Emotions

In this section, we analyze which emotions benefit from personality information and post history using emotion-wise QWK. Figure 2 shows the QWK improvement for each emotion, averaged across the evaluated LLMs. The improvement is calculated by subtracting the 0-shot QWK from that of each personalization method. A positive value indicates improvement over the 0-shot setting, whereas a negative value indicates performance degradation. As shown in Figure 2, personality information and post history exhibited different patterns in the emotions they improved.

When only Personality60 was provided, performance decreased relative to the 0-shot setting for many emotions. In particular, the QWK for anger decreased by approximately 0.08 on average. In contrast, Big Five yielded slight improvements for sadness, anticipation, and trust, although the overall gains were small.

Recent, which uses post history, produced substantial improvements for anger, fear, and disgust. In particular, the QWK increased by approximately

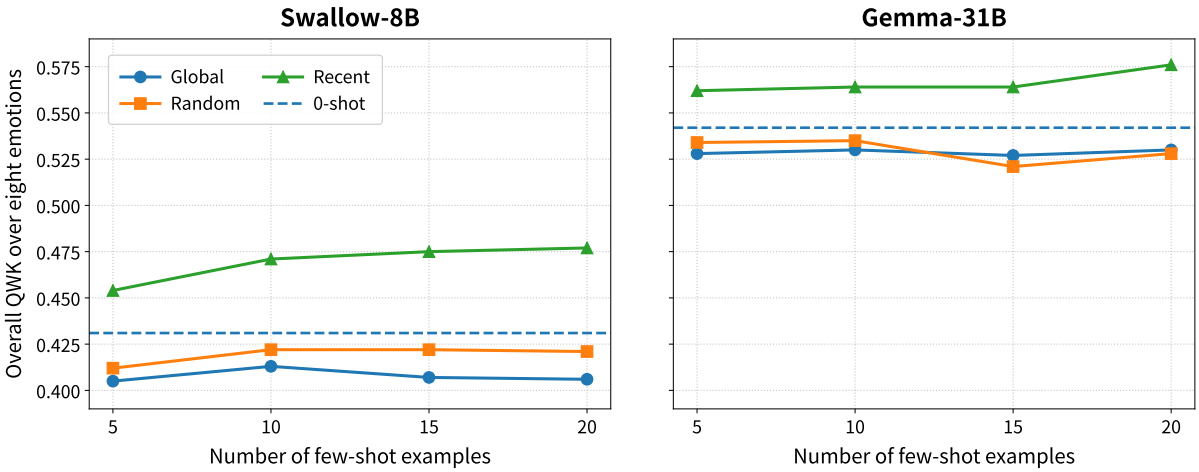


Figure 3: Changes in the Overall QWK across eight emotions with different numbers and selection strategies of few-shot examples. Global denotes training examples selected without considering the writer, Random denotes posts randomly selected from the same writer, and Recent denotes temporally close posts from the same writer that precede the input post. Dashed lines indicate the performance of the 0-shot setting for each model.

0.06 for fear, 0.06 for anger, and 0.05 for disgust. In contrast, no improvement was observed for anticipation or trust, indicating that the effectiveness of post history varies across emotions.

When personality information was combined with post history, the methods maintained large gains for anger, fear, and disgust, as with Recent alone, while further improving sadness. In particular, Big Five + Recent achieved the largest gain of approximately 0.08 for fear and approximately 0.06 for both anger and disgust. In contrast, the gain for trust was small, and some settings underperformed the 0-shot setting.

Overall, these results indicate that post history is particularly effective for estimating anger, fear, and disgust, for which writer-specific patterns of emotional expression are likely to be reflected. Although personality information alone provides only limited improvements, combining it with post history yields additional gains for some emotions, particularly sadness.

5.2 Effect of Few-shot Example Size

To analyze the effects of the number of examples and the example selection method on performance, we conducted experiments using Swallow-8B and Gemma-31B while varying the number of few-shot examples among 5, 10, 15, and 20. We considered three example selection methods: Global, which selects examples from the training data without distinguishing writers; Random, which randomly selects posts written by the same writer; and Recent, which selects temporally close posts written before the target post. The results are shown in Figure 3.

For Swallow-8B, Global and Random underperformed 0-shot at all example sizes, whereas Recent consistently improved performance from 0.454 with 5 examples to 0.477 with 20. This indicates that temporal relevance is more important than simply increasing the number of few-shot examples.

For Gemma-31B, Global and Random also underperformed 0-shot, while Recent consistently surpassed it. Its performance remained stable from 5 to 15 examples and increased to 0.576 with 20, suggesting model-dependent gains from additional history.

Overall, these results show that few-shot performance depends more on example relevance than quantity. Temporally close posts from the same writer outperformed 0-shot even with few examples and provided further gains as the number increased.

6 Conclusion

We addressed writers' subjective emotion intensity estimation and proposed a personalized method that incorporates personality information and post history into the LLM prompt. Experiments on WRIME showed that temporally close post history consistently improved performance, whereas personality information alone had limited effects. Combining both yielded further gains, demonstrating the effectiveness of prompt-based personalization for LLM emotion analysis.

Acknowledgments

This work was supported by JST BOOST Program Japan Grant Number JPMJBY24036821.

Limitations

Our experiments are limited to the WRIME corpus, which contains Japanese SNS posts, and the generalizability of the findings to other languages and domains remains to be investigated. In addition, the available post history is limited, and the effectiveness of the proposed method may depend on the number and temporal distribution of posts available for each writer. Furthermore, our method assumes that emotion labels for past posts are available at prediction time. In practical applications, these labels may need to be collected through self-reports or estimated automatically.

References

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural Sentiment Classification with User and Product Attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659.

Jacob Cohen. 1968. Weighted Kappa: Nominal Scale Agreement Provision for Scaled Disagreement or Partial Credit. *Psychological Bulletin*, 70(4):213–220.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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*, pages 4171–4186.

Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. Continual Pre-Training for Cross-Lingual LLM Adaptation: Enhancing Japanese Language Capabilities. In *First Conference on Language Modeling*.

Gemma Team. 2026. Gemma 4 Technical Report. *arXiv:2607.02770*.

Lewis R. Goldberg. 1992. The Development of Markers for the Big-Five Factor Structure. *Psychological Assessment*, 4(1):26–42.

Tomoyuki Kajiwara, Chenhui Chu, Noriko Takemura, Yuta Nakashima, and Hajime Nagahara. 2021. WRIME: A New Dataset for Emotional Intensity Estimation with Subjective and Objective Annotations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2095–2104.

LLM-jp. 2024. LLM-jp: A cross-organizational project for the research and development of fully open Japanese LLMs. *arXiv:2407.03963*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of the Seventh International Conference on Learning Representations*.

Chenyang Lyu, Linyi Yang, Yue Zhang, Yvette Graham, and Jennifer Foster. 2023. Exploiting Rich Textual User-Product Context for Improving Personalized Sentiment Analysis. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1419–1429.

Youmi Ma, Sakae Mizuki, Kazuki Fujii, Taishi Nakamura, Masanari Ohi, Hinari Shimada, Taihei Shiotani, Koshiro Saito, Koki Maeda, Kakeru Hattori, Takumi Okamoto, Shigeki Ishida, Rio Yokota, Hiroya Takamura, and Naoaki Okazaki. 2025. Building instruction-tuning datasets from human-written instructions with open-weight large language models. In *Proceedings of the Second Conference on Language Modeling*.

Piotr Milkowski, Marcin Gruza, Kamil Kanclerz, Przemyslaw Kazienko, Damian Grimling, and Jan Kocon. 2021. Personal Bias in Prediction of Emotions Elicited by Textual Opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 248–259.

Robert Plutchik. 1980. A General Psychoevolutionary Theory of Emotion. *Theories of Emotion*, pages 3–33.

Qwen Team. 2026. Qwen3.6-27B: Flagship-level coding in a 27B dense model.

Takako Saito, Tomoyasu Nakamura, Toshihiko Endo, and Madoka Yokoyama. 2001. Standardization of Big Five Scales Using the Adjective Check List. *Kyushu University Psychological Research*, 2:135–144.

Haruya Suzuki, Yuto Miyauchi, Kazuki Akiyama, Tomoyuki Kajiwara, Takashi Ninomiya, Noriko Takemura, Yuta Nakashima, and Hajime Nagahara. 2022a. A Japanese Dataset for Subjective and Objective Sentiment Polarity Classification in Micro Blog Domain. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7022–7028.

Haruya Suzuki, Sora Tarumoto, Tomoyuki Kajiwara, Takashi Ninomiya, Yuta Nakashima, and Hajime Nagahara. 2022b. Emotional Intensity Estimation Based on Writer's Personality. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 1–7.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning Semantic Representations of Users and Products for

Document Level Sentiment Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1014–1023.

Zhen Wu, Xin-Yu Dai, Cunyan Yin, Shujian Huang, and Jiajun Chen. 2018. Improving review representations with user attention and product attention for sentiment classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, pages 5989–5996.

You Zhang, Jin Wang, Liang-Chih Yu, Dan Xu, and Xuejie Zhang. 2024. Improving Personalized Sentiment Representation with Knowledge-Enhanced and Parameter-Efficient Layer Normalization. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pages 8877–8889.

Predicting Steering Vectors and Adapter Weights for Few-Shot Author-Style Transfer

Leonard Popp **Danni Liu** **Supriti Sinhamahapatra** **Jan Niehues**
Karlsruhe Institute of Technology, Germany
leonard.popp@student.kit.edu, {firstname.lastname}@kit.edu

Abstract

Adapting large language models to an individual author's style from a few examples is challenging, and scientific writing sharpens the difficulty: formal conventions leave little surface variation, and authors write about their own topics, so extracted "style" easily entangles with content. We study style-conditioned abstract generation from a few example abstracts per author and propose three methods: (1) contrastive activation steering, (2) a network that predicts steering vectors, and (3) a hypernetwork that predicts LoRA adapters. We find a consistent trade-off between style imitation and output quality: fine-tuning buys most of the available style signal but forfeits fluency, while the hypernetwork achieves the best trade-off on both seen and unseen authors. Our steering operates at author level, contrasting an author's abstracts against style-neutral generations for the same content. This holds topic fixed, removes the need for a predefined style inventory, and outperforms inventory-based steering. Moreover, our analyses demonstrate that manually extracted and predicted steering vectors are near-orthogonal yet score comparably, indicating that style conditioning here can admit at least two unrelated directions rather than requiring one particular axis.

1 Introduction

By in-context learning (Brown et al., 2020), large language models can solve a wide range of tasks by conditioning on a few demonstrations without parameter updates. Author-level style control, however, is not reliably one of these tasks. While coarse style attributes such as sentiment or emotion can be controlled through activation-level interventions (Subramani et al., 2022; Rimsky et al., 2024; Konen et al., 2024; Zhang et al., 2025b), imitating an individual author from a handful of texts is harder: Horvitz et al. (2024) find that an 800M model fine-tuned on authorship embeddings outperforms GPT-4 at few-shot transfer toward specific users. This is the *cold-start* setting of user-level personalization (Zhang et al., 2025c): per-user fine-tuning (Tan et al., 2024c; Dan et al., 2025) requires a history the new author does not have, retrieval over that history is equally unavailable (Salemi et al., 2024; Richardson et al., 2023), and a few examples are often insufficient for the model to extract style (Reif et al., 2022; Patel et al., 2022). Scientific writing exacerbates the difficulty. Formal conventions leave little surface variation, and authors write in their style about their topics, so a model extracting an author's style risks copying content rather than form (Wegmann et al., 2022; Patel et al., 2025).

Existing approaches address parts of this problem but leave a clear gap. Activation steering has been evaluated on coarse stylistic distinctions (Konen et al., 2024) or in settings with substantial per-attribute supervision (Rimsky et al., 2024; Tan et al., 2024a), and StyleVector (Zhang et al., 2025b) computes and stores one contrastive vector per user offline. Adapters generated by hypernetworks (Ha et al., 2016) remove per-user training, but Text-to-LoRA (Charakorn et al., 2025) conditions on natural-language task descriptions, and P2P (Tan et al., 2026) on user profiles assembled from history summaries and retrieval. None of these works targets fine-grained, author-level style in a strict few-shot regime. We study style-conditioned abstract generation from three example abstracts per author, comparing three baselines (zero-shot, few-shot, LoRA) against three proposed methods (§4): 1) contrastive activation steering, 2) a network predicting steering vectors, and 3) a hypernetwork predicting LoRA adapters. Our contributions are:

- We expose a consistent trade-off between style imitation and output quality across all six methods, on which the hypernetwork appears to occupy the most efficient position (§6.4).
- We show that manually computed and predicted steering vectors are near-orthogonal yet yield

Proceedings of the 11th Workshop on Natural User-generated Text, pages 227–243
October 28, 2026 ©2026 Association for Computational Linguistics

comparable style accuracy, so at least two unrelated directions support style conditioning and no particular axis is required (§7.1).
- We adapt contrastive steering to the author level by contrasting ground-truth abstracts against style-neutral model generations for the same content, and find it matches or exceeds the method of Konen et al. (2024) (§7.2).

2 Related Work

Few-Shot Personalization of LLMs Style-conditioned generation sits at the intersection of text style transfer and user-level personalization (Zhang et al., 2025c). Horvitz et al. (2024) target the low-data regime by conditioning a small model on authorship embeddings for few-shot transfer toward individual users. The strongest personalization results, however, come from per-user fine-tuning that trains dedicated LoRA adapters (Hu et al., 2022) for each user (Tan et al., 2024c,b; Dan et al., 2025). This scales poorly in the number of users and degrades when user histories are limited. This work targets the cold-start setting at author-level granularity, in scientific writing, a domain whose formal constraints leave comparatively little surface variation to exploit.

Activation Steering Hidden activations contain latent directions that control stylistic attributes (Subramani et al., 2022; Zhang et al., 2025a). Konen et al. (2024) derive style vectors by contrasting mean activations of a target style against a fixed set of remaining styles, at a coarse granularity such as modern versus old English. We instead contrast an author's ground-truth abstracts against style-neutral abstracts the model generates for the same content, which removes the need for a predefined inventory, holds topic constant, and operates at author level. The matched-content contrast is close in spirit to Zhang et al. (2025b); our contribution is its use at author level in scientific writing and, primarily, the prediction of author-specific interventions from three examples. Zhang et al. (2025b) store one contrastive vector per user, computed offline from that user's texts. We further train a network to predict steering vectors in a single forward pass from three examples, so no per-user artifact is computed or stored.

Hypernetwork-Generated Adapters Hypernetworks (Ha et al., 2016) generate adaptation parameters directly, removing the need for per-user training: prior work produces attribute-conditioned prefixes for a frozen LM (Clive et al., 2022) and user-specific LoRA adapters for perspectivist classification (Ignatev et al., 2025). Text-to-LoRA (Charakorn et al., 2025), whose backbone we adopt, maps a natural-language task description to LoRA weights for zero-shot adaptation to unseen tasks. We instead predict style-conditioned adapters for a fixed task from an aggregated embedding of examples. Closest to our setting, Tan et al. (2026) map an encoded user profile to LoRA weights for long-form generation, but rely on an LLM-generated history summary and a retrieval pipeline, whereas we condition on three abstracts alone.

3 Task and Evaluation

Task Formulation Given a target author a, a reference set $R_a = \{r_1, r_2, r_3\}$ of three abstracts by a, and a *structured outline* s of an unseen paper, the task is to generate an abstract $\hat{y} = M(s, R_a)$ that describes s in the style of a. Outlines are produced from the paper's LaTeX source (§5.1); no model observes the original abstract as input. The base model M is frozen throughout, and no method requires per-author training at deployment. The small size of R_a is a property of the cold-start setting (Zhang et al., 2025c) rather than a hyperparameter. Unlike prior work, we assume neither a rich user profile nor a retrieval corpus (Tan et al., 2026). We evaluate on authors seen during training and on held-out authors entirely (§5.1).

Author Style Evaluation We measure style by authorship attribution. Instead of a standard classifier which fixes the label set at training time, our unseen-author regime requires attributing text to new authors introduced at inference from three examples. Therefore, we adopt k-NN retrieval in a learned embedding space, which admits new authors without retraining. Specifically, an encoder is first fine-tuned on authorship classification. Then we discard its head, and create document embeddings by mean-pooling With the example abstracts forming the reference database, a query is assigned the majority label among its k nearest neighbors under cosine distance, falling back to the nearest neighbor when no majority exists. We report accuracy and mean reciprocal rank (MRR), the latter being continuous and therefore less sensitive to sampling variance. Note that style and content are not fully separable in this metric, as abstracts generated without author information are attributed well

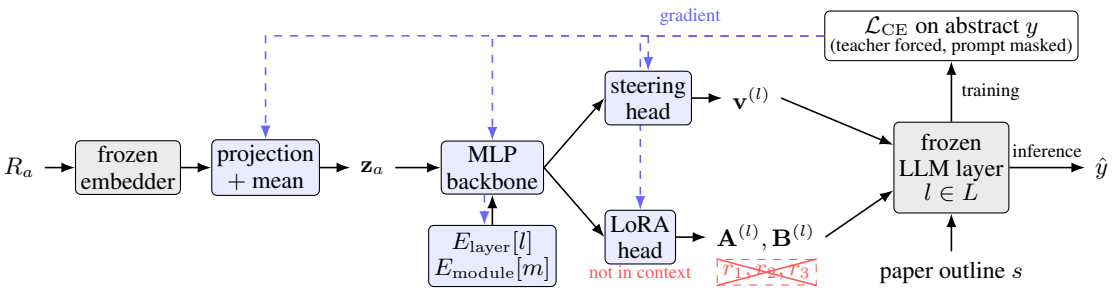


Figure 1: Both style control networks differ only in their output head. The author-specific example abstracts R_a are embedded into $\mathbf{z}_a$, concatenated with a layer embedding (for the hypernetwork, also a module embedding) and passed through the MLP backbone. The steering head emits a vector added to the residual stream. The LoRA head emits adapter weights inserted into the attention projections. The network is queried once per target layer $l \in L$ (and per module m for LoRA prediction).

above chance (§6.1), a known issue of authorship representations (Wegmann et al., 2022; Huertas-Tato et al., 2024). So we only use style accuracy to compare models rather than as an absolute metric.

Topic Preservation Evaluation The generated abstracts must not deviate from the main text's topic. Therefore, we detect off-topic generation with a binary LLM-judge. `GPT-OSS-20B` (OpenAI, 2025) receives an abstract and the introduction of a paper and decides whether both originate from the same document (prompt in Appendix A.1).

Text Quality Evaluation Besides topic preservation, we evaluate the intrinsic quality of the generated abstracts. Perplexity is an established metric for text fluency (Holtzman et al., 2020) but is untrustworthy here, as it rewards the predictable, repetitive text that aggressive style conditioning may produce. Instead, we measure quality by pairwise preference by letting `GPT-OSS-20B` compare each generated abstract against the author's real abstract for the same paper (prompt in Appendix A.2). Each pair is judged twice in swapped order to cancel position bias (Ko et al., 2020; Qin et al., 2024). We report the preference rate against the human-written abstract. Since our corpus predates 2022 when LLM-assisted writing emerged (§5.1) and the judge tends to favor LLM-generated content (Zheng et al., 2023), preference rates are high across all models. However, note that it is the differences between the preference rates, not their absolute level, that carry the information.

4 Style Control Methods

Manual Contrastive Steering Hidden activations contain directions that control stylistic attributes, and adding a style vector to selected layers steers generation toward the corresponding style. Prior work (Konen et al., 2024) obtains the steering vector by contrasting a target style against the mean activation of all other styles in a fixed inventory. This does not transfer to our case, as such an inventory hardly exists in practice, and contrasting one author against others writing about different topics leaves content in the vector. Instead, we contrast each reference abstract against the model's own "neutral" abstract for the same paper, generated zero-shot from s_i in the model's default style. Specifically, for layer l, the steering vector is

$$\mathbf{v}_a^{(l)} = \bar{\mathbf{a}}_{\text{auth}}^{(l)} - \bar{\mathbf{a}}_{\text{gen}}^{(l)}, \tag{1}$$

where both terms are activations mean-aggregated over tokens. The three per-reference vectors are averaged and injected during generation,

$$\mathbf{a}_{\text{steered}}^{(l)}(x) = \mathbf{a}^{(l)}(x) + \lambda\, \mathbf{v}_a^{(l)}, \tag{2}$$

with intensity λ and target layers L selected on validation (§5.3). Our approach outperforms the inventory-based method of Konen et al. (2024), as shown in §7.2. Three constraints remain. First, the contrast requires one neutral generation per reference, which in turn requires the outline of the corresponding reference paper. Manual steering therefore consumes strictly more than R_a and does not satisfy the deployment setting of §3; we report it not as a competing method under that setting, but as a reference point the two learned methods must match using the three abstracts alone. Second, it requires a λ search per setting: we sweep $\lambda \in \langle \text{grid} \rangle$ on validation and select $\lambda = 0.875$ for the test configuration, whereas the layer comparison in Table 5 was run at $\lambda = 1.0$ before this search. Third, the intervention is applied to the residual stream

at the ⟨output of each target block / post-attention position⟩ during both prefill and autoregressive decoding, and the activations in Equation 1 are mean-pooled over the ⟨abstract tokens only, with prompt tokens excluded⟩.

Learned Steering We tackle both constraints by predicting the intervention directly from R_a. Each reference is embedded with a frozen STAR encoder[1] (Huertas-Tato et al., 2024), then projected and averaged into an author representation

$$\mathbf{z}_a = \frac{1}{|R_a|} \sum_i \text{Proj}\big(\text{encoder}(r_i)\big). \tag{3}$$

Concatenating $\mathbf{z}_a$ with a learned layer embedding forms the input passed to an MLP backbone with residual blocks following Charakorn et al. (2025) (Figure 1; details in Appendix B). A linear head projects to the hidden dimension of M and its output is added to the residual stream, $\mathbf{a}^{(l)}_{\text{steered}}(x) = \mathbf{a}^{(l)}(x) + \mathbf{v}^{(l)}_a$. The head is zero-initialized, so the intervention is a no-op at the start of training. No explicit λ is applied, as the network learns per-layer magnitudes. Despite this flexibility, a downside is that a single vector per layer shifts the residual stream uniformly for every token, independently of what is being written. We tackle this next.

Hypernetwork for LoRA Weights Unlike steering vectors, LoRA adapters condition their update on the layer input x rather than adding a constant offset, so the intervention varies with what the model is writing. Therefore, we extend the previous setup to predict LoRA weights, inspired by Charakorn et al. (2025) which predicts LoRA weights given novel task descriptions. Here, we adopt their architecture, with a module embedding for $m \in \{\texttt{q\_proj}, \texttt{v\_proj}\}$ concatenated alongside the layer embedding, and four linear heads emit the flattened adapters $\mathbf{A}^{(l)} \in \mathbb{R}^{r \times d_{\text{in}}}$ and $\mathbf{B}^{(l)} \in \mathbb{R}^{d_{\text{out}} \times r}$, where r is the LoRA rank. This modifies the forward pass as

$$\mathbf{a}^{(l)}(x) = \mathbf{W}^{(l)}_0 x + \mathbf{B}^{(l)}\mathbf{A}^{(l)}x. \tag{4}$$

The module initialization mirrors standard LoRA, where the $\mathbf{A}$-heads are initialized with zero weights and uniform biases, and the $\mathbf{B}$-heads with zero weights and biases. This leaves the adapters as a no-op at initialization and was essential for training stability in our case. We treat the set of adapted layers as a design choice, restricting it to a middle-layer subset as it acts as a regularizer and improves downstream metrics (Table 5).

Training As shown in Figure 1, both learned steering and the hypernetworks are trained end-to-end with the author's ground-truth abstract. The base model and the encoder stay frozen, and gradients pass through M into the prediction network alone. Crucially, M receives the zero-shot prompt: the references never enter its context, so style is carried entirely by the predicted parameters.

5 Experimental Setup

5.1 Data Creation

Source and Filtering Our dataset is built from an arXiv snapshot. We retain papers with permissive licenses[2] in the `cs.AI` and `cs.LG` categories. Publication year is capped at 2022, so that ground-truth abstracts cannot have been written by a modern LLM: the style targets are *guaranteed human*. Lacking unique author identifiers, we treat identical author names as one person and discard authors publishing in over five arXiv categories as a disambiguation heuristic. We assume the first author is a paper's primary writer. Papers without parsable LaTeX source or an introduction are excluded.

Outline Creation As defined in §3, the models receive an outline rather than the full paper. This ensures that minimal style information is available in the input. We clean the LaTeX source to plain text and prompt Qwen3-4B-Instruct-2507 to produce bullet points under four headings: Context, Methodology, Results, and Conclusion. Additional details and prompts are in Appendix A.3.

Statistics After filtering, the dataset contains 1,232 papers from 203 authors, averaging 6.07 papers each (range 5-14). One paper per author forms the validation set and one the test set, and the remaining 826 form the training set. Each entry pairs a paper's outline with its ground-truth abstract, its introduction, and three abstracts sampled randomly from the same author's remaining papers.

Unseen Author Test Set To evaluate generalization, we create a second test set where none of the authors appeared in training for either the generation models or the author classifier. In this set, the

[1] STAR is trained contrastively for authorship and therefore expected to be more stylistic than semantic. It also outperformed `Qwen/Qwen3-Embedding` in initial experiments.

[2] Some licenses require mentioning the authors. We include this in our repository (deanonymized upon publication).

authors have exactly four papers, which guarantees disjointness from the base dataset. It contains 1,312 papers from 328 authors: three per author serve as prompt examples and one as the test sample.

5.2 Baselines

We compare our methods to three standard ones, with the corresponding prompts in Appendix A.4.

Zero-Shot The base model generates an abstract from the outline alone. It receives no author information, and therefore provides a lower bound on style accuracy and a quality anchor.

Few-Shot The zero-shot prompt is extended with the three reference abstracts, and the model is instructed to extract and apply the author's style. This measures style transfer without parameter updates.

LoRA The base model is fine-tuned on the few-shot prompt with adapters on the query and value projections at rank $r = 8$, matching the hypernetwork's adapter budget so that the two are directly comparable. It represents the strongest style baseline. Note that here LoRA is *not per-author*, as our data only contain roughly five abstracts per author. Instead, a single set of LoRA adapters is optimized across all training authors and conditioned at inference on the three author-specific examples.

5.3 Implementation Details

All models use `Qwen3-4B-Instruct-2507` (Qwen Team, 2025) as the frozen base model M (36 layers, $d = 2{,}560$). We decode with sampling ($T = 0.7$, repetition penalty 1.05, top-p $= 0.8$, top-k $= 20$) and generate 1,024 tokens at most. The kNN author classifier is a fine-tuned `distilbert-base-uncased` with $k = 3$ and cosine distance. Both learned steering and the hypernetwork use a four-block MLP backbone with an author representation of 128 dimensions.

Configuration searches use the validation set, while the test set is reserved for final evaluation. Manual steering uses our contrast method at layers 18-20 with $\lambda = 0.875$. Learned steering encodes the three example abstracts with the STAR encoder (Huertas-Tato et al., 2024) and predicts a shift vector for all 36 layers. The hypernetwork uses the same STAR encoder and predicts rank-8 adapters for the query and value projections, matching the LoRA baseline's rank. Further implementation and hyperparameter details are in Appendix B.

	Pred. Positive	Pred. Negative	Total
GT Positive	1,216	16	1,232
GT Negative	2	1,230	1,232
Total	1,218	1,246	2,464

Table 1: Confusion matrix for the topic preservation judge. It attains 99.3% accuracy, misclassifying 18 of 2,464 abstract-introduction pairs. Negative pairs are constructed by randomly pairing abstracts with unrelated introductions.

	Content	Style	Style Acc. (%)
	Target author's papers		
(1)	target	human (ground truth)	53.2
(2)	target	LoRA (target author)	49.8
(3)	target	none (zero-shot)	31.0
	Different author's papers		
(4)	foreign	LoRA (target author)	0.3
(5)	foreign	none (zero-shot)	0.0
(6)	Random guessing		0.5

Table 2: Evaluation of the k-NN author classifier. Each row is a set of abstracts labeled with the target author, varying the paper (*Content*) and the source of writing style (*Style*). Removing style at fixed content (row 3) costs 22.2 points, yet the remaining 31.0% is attainable from content alone. Conversely, styling a different author's paper (rows 4-5) falls to chance.

6 Quantitative Results

6.1 Metric Validation

Before reporting the main results, we validate the two automatic metrics: the topic preservation judge and the author classifier.

Topic Preservation Judge We first verify that `GPT-OSS-20B` can reliably detect whether a generated abstract describes the intended paper. As shown in Table 1, on an evaluation set of 2,464 abstract-introduction pairs, the model achieves 99.3% accuracy, with 16 false negatives and 2 false positives. Therefore, we adopt it as a guard against off-topic generation.

Content Entangled with Style Table 2 shows results of evaluating the k-NN style classifier. First, the classifier attains 53.2% accuracy on held-out ground-truth abstracts against random guess accuracy of 0.5%. Two experiments establish what it measures. Removing style while holding content fixed, by classifying style-neutral abstracts generated from the same outlines, drops accuracy to 31.0%. The 22.2-point difference is what the classifier loses when author style is removed under this

intervention, and the remaining 31.0 is what content alone supplies; neither figure isolates a purely stylistic or purely content-driven component. This score stays far above chance, and as these abstracts preserve the paper's topic, the residual shows the classifier also draws on content, an entanglement consistent with known difficulties in authorship attribution (Huertas-Tato et al., 2024; Wegmann et al., 2022). Reversing the manipulation, by varying content while holding style conditioning fixed, sends every model to or below chance, equally consistent with a classifier that cannot recognize unfamiliar content and with generators that cannot carry style across topics.

Interpreting Style Accuracy The 53.2% ground-truth level reflects the classifier's attribution ability rather than a property of the generative models. More consequentially, the style-neutral condition above comes from the same zero-shot model that serves as our first baseline in §6.2, so its 31.0% is not a style score but the level attainable from content alone. We therefore read style accuracy relative to a content-only *reference point* of 31.0% and a classifier *ceiling* of 53.2%. Neither is a strict bound: the few-shot baseline falls below 31.0% (§6.2).

6.2 Results on Seen Authors

Table 3a reports test set results for our three approaches alongside the three baselines and the ground truth, evaluated on the 203 authors seen during training. We report preference rate as the primary quality metric and author classifier accuracy as the primary style metric. All models preserve the paper's topic at comparable levels (95.6-97.5%), which is a precondition for comparing style accuracy rather than a discriminating result.

Few-shot baseline fails to convey author style. The few-shot baseline scores below the zero-shot baseline on both style metrics (28.1% and 0.36 against 31.0% and 0.40), despite receiving the same examples the LoRA baseline is trained with. It thus falls below the content-only reference point, potentially because instruction-following pulls the output toward a generic instructed register, moving embeddings away from any specific author. Quality degrades as well (91.5% preference rate).

Direct fine-tuning maximizes style at the cost of quality. The LoRA setup attains the highest style accuracy at 49.8%, roughly 84% of the range above the content-only reference point. However, its preference rate falls drastically to 38.1% from 97.9% for the zero-shot baseline, so fine-tuning purchases nearly all of the available style signal by sacrificing what makes the abstracts usable. Manual inspection reveals fluency issues with frequent repetitions.

Model	Pref. Rate (%)	Topic Pres. (%)	Style Acc. (%)	MRR
Zero-shot	**97.9**	96.6	31.0	0.40
Few-shot	91.5	97.0	28.1	0.36
LoRA	38.1	95.6	**49.8**	**0.55**
Manual steering	94.3	97.5	34.5	0.42
Learned steering	89.6	96.1	35.0	0.43
Hypernet LoRA	91.3	97.0	36.5	0.46
Ground truth	–	97.0	53.2	0.61

(a) Seen authors test set ($n = 203$). Style accuracy is bounded below by 31.0, the level attainable from content alone (§6.1), and above by the classifier ceiling of 53.2.

Model	Pref. Rate (%)	Topic Pres. (%)	Style Acc. (%)	MRR
Zero-shot	**97.4**	99.1	22.9	0.31
Few-shot	90.2	99.7	18.3	0.26
LoRA	39.0	97.0	**37.5**	**0.47**
Manual steering	95.1	98.5	26.2	0.37
Learned steering	87.5	99.4	28.4	0.38
Hypernet LoRA	95.5	99.4	31.4	0.40
Ground truth	–	100.0	41.5	0.49

(b) Unseen authors test set ($n = 328$). Neither the models nor the author classifier were trained on any sample from this set. Style accuracy is bounded below by 22.9, and above by the classifier ceiling of 41.5.

Table 3: Style and quality results on two tests sets with (a) seen and (b) unseen authors.

Activation steering preserves quality but yields small gains. Both steering approaches improve style while leaving quality largely intact. Manual steering reaches 34.5% at a preference rate of 94.3%, the highest of any model improving on the zero-shot style accuracy, and the prediction network reaches 35.0% at 89.6%. Predicting vectors therefore matches hand-computed ones on style without a neutral generation per reference or a search over intensity. The two differ by a single abstract, and §7.1 shows they achieve this by steering in nearly orthogonal directions.

Hypernet LoRA balances style and quality. The hypernetwork LoRA reaches 36.5%, roughly a quarter of the range, while retaining a preference rate of 91.3% that direct fine-tuning forfeits. Both amortize across authors rather than fitting each sep-

arately, differing in that LoRA carries style through the context window with a shared adapter, whereas the hypernetwork carries it through adapter weights predicted per author. Note the two methods are matched on adapter budget but not on information: LoRA reads the example text directly and can copy surface phrasing, while the hypernetwork conditions only on their embeddings.

6.3 Generalization to Unseen Authors

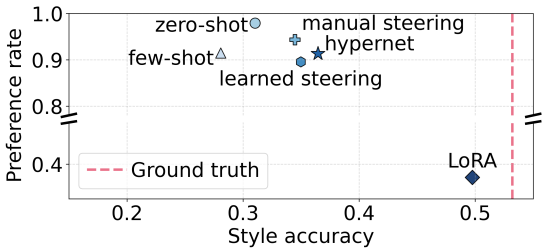
So far, the models were tested on authors present in the training data. We now assess if style conditioning transfers to authors that neither the models nor the author classifier have seen (Table 3b).

Model rankings are preserved. Every value is lower than on the seen authors, including the ground truth (41.5% against 53.2%). Only the relative ordering is informative, and it is identical, with few-shot again last and below the zero-shot baseline. As the seen and unseen sets share no authors, generation noise is an implausible explanation for the few-shot result.

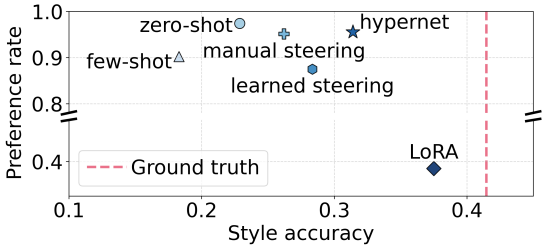
Hypernet retains position. The hypernetwork improves style by 8.5 points over the zero-shot baseline while giving up 1.9 points of preference rate, ranking second on all three metrics, and the standard LoRA setup again buys the highest style accuracy at a preference rate of 39.0%. The trade-off structure thus reappears unchanged on authors no component of the system was trained on. The hypernetwork now recovers close to half of the range against roughly 79% for LoRA, where the seen-author fractions were a quarter and 84%, though the narrower unseen range inflates these fractions and they are not directly comparable across sets.

6.4 Style-Quality Trade-Off

Figure 2 plots preference rate against style accuracy. We characterize the trade-off by Pareto dominance on point estimates. Three models are Pareto-optimal: the zero-shot baseline on preference rate, LoRA on style accuracy, and the hypernetwork LoRA between them. On the seen authors the frontier also contains manual steering, so the hypernetwork is the only one of our approaches Pareto-optimal on *both* sets at the configurations of Table 3a. Two qualifications apply. First, the two learned methods are reported at different intervention scopes: learned steering over all 36 layers and the hypernetwork over layers 18–20. At the common scope 18–20 (Table 5), the hypernetwork



(a) Seen author test set



(b) Unseen author test set

Figure 2: Style-quality trade-off by plotting the preference rate against style accuracy for each method, on seen (a) and unseen authors (b). The hypernetwork sits immediately before quality degrades.

retains higher style accuracy (36.5% vs. 34.0%) but learned steering attains the higher preference rate (97.5% vs. 91.3%), so neither dominates and the seen-set frontier is scope-dependent. Second, several style-accuracy differences correspond to few examples; manual and predicted steering differ by a single abstract on the seen-author test set. We therefore report paired bootstrap 95% confidence intervals in Table 3a and read Pareto statements as descriptive of point estimates rather than as significance claims. Relative to the hypernetwork, on the unseen set, LoRA buys 6.1 points of style accuracy for 56.5 points of preference rate. In contrast, the hypernetwork occupies the frontier between the two, retaining quality while recovering a meaningful share of the style range.

7 Analyses

In addition to quantitative results above, we conduct several analyses.

7.1 No Single Style Axis

As shown previously in Table 3a, manual steering and the activation vector prediction network reach comparable style accuracies. Since both add a vector to the model's hidden activations, we ask: do both methods achieve the same *behavior* with the same *internal mechanisms*? Table 4 compares the vectors both produce at layers 18 to 20, averaged

	L2 Norm			L1 / L2			Cosine Similarity
Layer	Manual	Pred.	Hidden	Manual	Pred.	Hidden	Manual vs. Pred.
18	13.4	14.3	39.9	17.3	40.1	11.8	−0.03
19	13.6	8.2	46.4	18.1	39.5	12.1	−0.05
20	15.1	12.2	49.4	19.4	39.7	12.0	−0.04

Table 4: Per-layer statistics for manual and predicted steering vectors, averaged over the 203 validation samples. *Hidden* indicates the *unsteered* activations the vectors are added to. L1/L2 measures how broadly a vector spreads its mass, with 40.4 expected of a dense Gaussian vector in 2,560 dimensions and 50.6 of a uniform one. The final column reports the average cosine similarity between the manual and predicted vector at each layer.

over 203 validation samples.

Manual and learned steering vectors are near-orthogonal. As shown in the last column of Table 4, the average cosine similarity between the manual and the predicted vectors is near zero at every layer (-0.03, -0.05, -0.04). In 2,560 dimensions this is the value expected of unrelated vectors, suggesting that the two sets of vectors are therefore *nearly orthogonal*. The near-orthogonality combined with their equivalent style accuracy shows that the predictor does not recover the manually extracted direction, and that at least two unrelated directions reach comparable style accuracy. This does not rule out the existence of a privileged direction; it shows that reaching one is not necessary. Establishing the stronger claim would require comparing vectors extracted from disjoint reference subsets of the same author against vectors from different authors, which we leave to future work.

Manual and learned steering vectors have similar magnitudes but different density. Across the two methods, the magnitude of the resulting perturbation is comparable. As shown in Table 4, both produce vectors whose L2 norm falls between 18% and 36% of the corresponding hidden activation norm, which suggests that a similar quantity of perturbation is required to influence the model's output regardless of the direction chosen. Despite comparable magnitudes, their *internal structure* does not match. The "L1/L2" column in Table 4 measures how broadly a vector spreads its mass. For the predicted vectors it sits at 39.5 to 40.1 at every layer, matching the value expected of a dense Gaussian vector in 2,560 dimensions. In contrast, the manual vectors reach only 17.3 to 19.4. The predicted vectors are therefore *dense*, spreading a small perturbation across many dimensions, whereas the manual vectors are *sparse*, concentrating the perturbation in a few dominant ones.

Layers	Style Acc. (%)	MRR	Pref. Rate (%)
Manual steering (validation)			
18–20, ours	36.0	–	–
20–22, ours	35.0	–	–
all 36, ours	2.5	–	–
18–20, Konen et al. (2024)	34.0	–	–
20–22, Konen et al. (2024)	31.0	–	–
Learned steering (test)			
18–20	34.0	0.44	**97.5**
all 36	**35.0**	0.43	89.6
Hypernetwork LoRA (test)			
18–20	**36.5**	**0.46**	**91.3**
all 36	34.5	0.43	71.4

Table 5: Effect of layer selection. Manual steering is reported on the validation set at $\lambda = 1.0$, where the configuration search measured style accuracy only (hence "–" for MRR and Pref. Rate). The learned interventions are reported on the test set.

7.2 Layer Selection

Table 5 shows the impact of which layers of the base model our interventions act on.

Manual steering requires a narrow band. Holding the contrast method fixed, layers 18-20 give the highest style accuracy (36.0%), one point above layers 20-22 (35.0%). The same ordering holds for the contrast of Konen et al. (2024) (34.0% vs. 31.0%), which our construction outperforms at both scopes. Applying the vector to all 36 layers, however, collapses style accuracy to 2.5%, far below the level attainable from content alone (§6.1). A likely explanation is accumulation along the residual stream. At the layers where we measure it, a single style vector displaces the hidden state by roughly a third of its norm (§7.1). Injecting one at every layer therefore compounds the displacement with depth, and risks driving the state out of the distribution the model was trained on. That the surviving band is a middle one matches the findings of Konen et al. (2024), who locate the

same band in a 32-layer model, and evidence that middle layers carry the most abstract representations (Wendler et al., 2024; Liu and Niehues, 2025). Here, mid-depth likely strikes the best balance, as an early perturbation is reshaped by its subsequent layers, while a late one leaves too few layers to realize it in surface form.

Hypernet prefers a narrow band but survives full depth. Compared to predicting adapters for all layers, restricting to layers 18-20 yields higher style accuracy (36.5% vs. 34.5%), MRR (0.46 vs. 0.43), and preference rate (91.3% vs. 71.4%). We evaluated only these two scopes, so the comparison shows that a narrow band beats full depth, not that 18-20 is optimal for the hypernetwork. The failure at full depth is milder than for manual steering. A plausible reason is learning the intervention end-to-end: Manual steering scales every layer by one global λ, while the hypernetwork predicts each layer's adapter separately, leaving it free to intervene weakly where a layer tolerates less perturbation, as the steering network measurably does (§7.1). That a narrow band wins here suggests this freedom is not sufficient on its own. Restricting the scope likely acts as a regularizer.

Localization is a property of the intervention. Learned steering makes the point more sharply still. At all 36 layers it reaches 35.0% style accuracy at an 89.6% preference rate, the same scope at which the manual vectors collapse to 2.5% on validation. The scope is identical; the vectors are not. Its behavior across scope also differs from the hypernetwork's: style accuracy is essentially flat (34.0% at layers 18–20 vs. 35.0% at full depth, within a point), while the narrow band instead buys quality, raising the preference rate to 97.5% from 89.6%. Which layers matter, and if narrowing them trades style for quality or improves both, is thus a property of the specific intervention rather than a fixed rule.

8 Conclusion

We studied author-level style conditioning in a strict cold-start setting: three abstracts, no history, no retrieval, and a frozen base model. Our results lead to three practical recommendations. (1) Style conditioning here is not a problem of finding the best intervention, but of navigating a trade-off each intervention pays. (2) The near-orthogonality of manual and learned steering vectors argues against two practices: extraction is not a search for *the* style direction, since at least two unrelated directions work equally well here and refining the contrast set may only tune an arbitrary choice; and vector similarity cannot validate a method when orthogonal vectors work equally well, only downstream metrics can. (3) In a contrastive steering vector, the negative should carry the style the output must deviate from, here the model's own default on the same content.

Limitations

Our style metric is entangled with content: style-neutral generations are attributed well above chance, so we can only read accuracy relative to an empirical reference point and ceiling, and comparisons are ordinal rather than absolute. The probe that varies content while fixing style conditioning lands at or below chance, which we cannot attribute to the classifier or to the generators. Readers should therefore treat every style number as a relative ranking within our setup, never as a measure of how well an author is imitated; disentangling the two would require a classifier trained to be content-invariant and a probe that isolates the generator from the metric.

Our main results do not by themselves establish that the learned interventions are author-specific rather than a generic shift toward human-written abstracts. We report the absence of this control as a limitation: without it, the improvement over zero-shot cannot be separated from a generic shift toward human-written abstracts, and the within- versus between-author similarity of predicted vectors and of the effective update $\mathbf{B}^{(l)}\mathbf{A}^{(l)}$ remains unmeasured.

The learned steering and hypernetwork was evaluated at two scopes, so we can only establish that a narrow band beats full depth, not that our selected band is optimal. Our layer choices should be read as a working configuration rather than a tuned one. A full sweep over bands and widths per method is left to future work, and the gap between methods may narrow once each is given its own best scope.

Our methods are run on one base model (Qwen3-4B-Instruct), one broad domain (pre-2022 `cs.AI/cs.LG` arXiv abstracts), and only English. Whether the trade-off structure is a property of style conditioning or of this configuration remains open. Replication across model scales, families, domains, and other languages is needed before generalizing.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. 2025. Text-to-loRA: Instant transformer adaption. In *Forty-second International Conference on Machine Learning*.

Jordan Clive, Kris Cao, and Marek Rei. 2022. Control prefixes for parameter-efficient text generation. In *Proceedings of the Second Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 363–382, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Yuhao Dan, Jie Zhou, Qin Chen, Junfeng Tian, and Liang He. 2025. P-react: Synthesizing topic-adaptive reactions of personality traits via mixture of specialized LoRA experts. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6342–6362, Vienna, Austria. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2026. The faiss library. *IEEE Transactions on Big Data*, 12(2):346–361.

David Ha, Andrew M. Dai, and Quoc V. Le. 2016. Hypernetworks. *CoRR*, abs/1609.09106.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Zachary Horvitz, Ajay Patel, Kanishk Singh, Chris Callison-Burch, Kathleen McKeown, and Zhou Yu. 2024. TinyStyler: Efficient few-shot text style transfer with authorship embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13376–13390, Miami, Florida, USA. Association for Computational Linguistics.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Javier Huertas-Tato, Alejandro Martín, and David Camacho. 2024. Understanding writing style in social media with a supervised contrastively pre-trained transformer. *Know.-Based Syst.*, 296(C).

Daniil Ignatev, Denis Paperno, and Massimo Poesio. 2025. Hypernetworks for perspectivist adaptation. In *Proceedings of the The 4th Workshop on Perspectivist Approaches to NLP*, pages 111–122, Suzhou, China. Association for Computational Linguistics.

Neel Jain, Ping yeh Chiang, Yuxin Wen, John Kirchenbauer, Hong-Min Chu, Gowthami Somepalli, Brian R. Bartoldson, Bhavya Kailkhura, Avi Schwarzschild, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. NEFTune: Noisy embeddings improve instruction finetuning. In *The Twelfth International Conference on Learning Representations*.

Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1109–1121, Online. Association for Computational Linguistics.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. 2024. Style vectors for steering generative large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 782–802, St. Julian's, Malta. Association for Computational Linguistics.

Danni Liu and Jan Niehues. 2025. Middle-layer representation alignment for cross-lingual transfer in fine-tuned LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15979–15996, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2025. gpt-oss-120b gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

Ajay Patel, Nicholas Andrews, and Chris Callison-Burch. 2022. Low-resource authorship style transfer with in-context learning. *CoRR*, abs/2212.08986.

Ajay Patel, Jiacheng Zhu, Justin Qiu, Zachary Horvitz, Marianna Apidianaki, Kathleen McKeown, and Chris Callison-Burch. 2025. StyleDistance: Stronger content-independent style embeddings with synthetic parallel examples. 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)*, pages 8662–8685, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518, Mexico City, Mexico. Association for Computational Linguistics.

Qwen Team. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. Association for Computational Linguistics.

Christopher Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *CoRR*, abs/2310.20081.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.

Daniel Tan, David Chanin, Aengus Lynch, Brooks Paige, Dimitrios Kanoulas, Adrià Garriga-Alonso, and Robert Kirk. 2024a. Analysing the generalisation and reliability of steering vectors. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024b. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6459–6475, Miami, Florida, USA. Association for Computational Linguistics.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024c. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6476–6491, Miami, Florida, USA. Association for Computational Linguistics.

Zhaoxuan Tan, Zixuan Zhang, Haoyang Wen, Zheng Li, Rongzhi Zhang, Pei Chen, Fengran Mo, Zheyuan Liu, Qingkai Zeng, Qingyu Yin, and Meng Jiang. 2026. Instant personalized large language model adaptation via hypernetwork. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23557–23580, San Diego, California, United States. Association for Computational Linguistics.

Anna Wegmann, Marijn Schraagen, and Dong Nguyen. 2022. Same author or just same topic? towards content-independent style representations. In *Proceedings of the 7th Workshop on Representation Learning for NLP*, pages 249–268, Dublin, Ireland. Association for Computational Linguistics.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Hanyu Zhang, Xiting Wang, Chengao Li, Xiang Ao, and Qing He. 2025a. Controlling large language models through concept activation vectors. In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 25851–25859. AAAI Press.

Jinghao Zhang, Yuting Liu, Wenjie Wang, Qiang Liu, Shu Wu, Liang Wang, and Tat-Seng Chua. 2025b. Personalized text generation with contrastive activation steering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7128–7141, Vienna, Austria. Association for Computational Linguistics.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2025c. Personalization of large language models: A survey. *Transactions on Machine Learning Research*. Survey Certification.

System Prompt

You are a strict data validation assistant. Your task is to compare an 'Introduction' and an 'Abstract' to determine if they describe the same specific subject matter, research topic, or underlying content.

Output Rules:
1. Return 'True' if the two texts describe the same core content or topic.
2. Return 'False' if the topics are unrelated, different, or contradictory.

Constraint: You must answer with EXACTLY one word: 'True' or 'False'. Do not provide explanations.

User Prompt

Please evaluate the following content pair:

Abstract
{abstract}

Introduction
{introduction}

Based on the content above, what is the determination (True/False)?

Figure 3: The prompt template used to query `GPT-OSS-20B` to classify the topic preservation of an abstract. `{abstract}` is replaced with the summary of the paper. `{introduction}` is replaced with the introduction the abstract is compared with.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Prompt Templates

A.1 Topic Preservation Judge Prompt

The topic preservation prompt (Figure 3) instructs `GPT-OSS-20B` to compare a generated abstract with the original paper introduction and decide whether they describe the same document, returning a single binary label. `GPT-OSS-20B` decodes greedily with the default (medium) reasoning effort.

A.2 Preference Rate Judge Prompt

The quality evaluation prompt (Figure 4) presents `GPT-OSS-20B` with two abstracts written for the same paper and asks which is of higher qualitative standard, based on clarity, scientific rigor, and narrative flow. `GPT-OSS-20B` decodes greedily with the default (medium) reasoning effort. For topic preservation the prompt is capped at 3,500 tokens; when a sample exceeds this, only the introduction is truncated from the end, which preserves the abstract in full since abstracts contain at most 242 words.

A.3 Outline Creation Prompt

Papers exceeding the context window are split into chunks of up to 7,750 tokens with a 250-token overlap, processed separately, and merged in a second pass. For papers whose plain text exceeds the model's context limit, each chunk is first processed individually using the prompt in Figure 5, which instructs the model to produce a compact bulleted outline of that chunk alone. These per-chunk outlines are then combined to form a single, structured outline using the prompt in Figure 6. For shorter papers that fit within the context window, the paper is directly summarized using the prompt in Figure 7 without chunking. In all cases, the model is a 4-bit quantized version of `Qwen3-4B-Instruct-2507`.

A.4 Abstract Generation Prompt

Two prompt templates are used for abstract generation, depending on whether the model receives author style examples. The zero-shot template (Figure 8) is used for the zero-shot baseline, the manual activation steering model, the steering vector prediction network and the hypernetwork model. It instructs the model to write a formal academic abstract from a structured paper outline, without any information about the target author. The few-shot template (Figure 9) is used for the few-shot baseline and the LoRA model. It extends the zero-shot template by providing three example abstracts by the target author, instructing the model to adopt that author's style while remaining faithful to the outline content.

B Implementation Details

Environment All models are loaded with Unsloth and Transformers (Wolf et al., 2020), and k-NN retrieval uses FAISS (Douze et al., 2026). Every experiment runs on a single NVIDIA TITAN RTX

System Prompt

You are an expert peer reviewer for a top–tier scientific journal. Your task is to compare two abstracts (Abstract A and Abstract B) written for the same paper and determine which one is of higher qualitative standard.

CRITERIA FOR THE WINNER:
Choose the abstract that a mainstream scientific audience would find more compelling, professional, and publishable. Focus on:
1. Clarity & Precision: Which text explains the problem, methodology, and results more clearly and with better scientific vocabulary?
2. Impact & Rigor: Which abstract sounds more objective, authoritative, and convincing?
3. Flow: Which abstract has a better narrative arc that makes the reader want to read the full paper?

OUTPUT FORMAT:
Respond strictly in valid JSON format:

```
{
  "reasoning": "1–2 sentences explaining why the winning abstract is more convincing to a reviewer.",
  "winner": "<A, B>"
}
```

User Prompt

Please evaluate the following two abstracts.

ABSTRACT A:
{abstract_a}

ABSTRACT B:
{abstract_b}

Figure 4: The prompt template used to query `GPT-OSS-20B` for the quality win rate evaluation. `{abstract_a}` is replaced with the generated abstract and `{abstract_b}` with the human-written ground-truth abstract. Each pair is evaluated twice with the input order swapped to eliminate positional bias.

System Prompt

You are a precise scientific summarizer.
Your goal: Synthesize the provided text chunk into a high–density, bulleted summary.
Constraints:
1. Max 400 words. Focus on maximum information density.
2. Preserve all key details important for writing an abstract or summary of the paper.
3. Remove redundancies and repetition.
4. Maximize information density: Prioritize factual content over linguistic fluency or reading flow.
5. Use a dry, compact, and technical style. Avoid flowery language or unnecessary transition words.
6. Especially do NOT use introductory filler (e.g. "This section discusses...").
7. Output ONLY the summary text.

User Prompt

Generate a comprehensive bulleted summary for this section:

{text_chunk}

Figure 5: The prompt template used to query `Qwen3-4B-Instruct-2507 4-bit` for chunk summarizations. `{text_chunk}` is replaced with a chunk of the paper.

System Prompt

You are a senior scientific editor.
Your goal: Synthesize provided informations into a high–density, bulleted summary that contains every detail necessary to write a formal abstract.

Constraints:
1. Length: 400–500 words. Focus on maximum information density.
2. Structure: Use exactly four bulleted sections: CONTEXT (problem/objective), METHODOLOGY (design/sample/tools), RESULTS (specific findings/data), and CONCLUSION (significance/implications).
3. Requirements: DO NOT omit specific metrics, sample sizes, or p–values. Ensure the logical thread from hypothesis to implication is intact.
4. Use bullet points only.
5. Format: PLAIN TEXT ONLY. No Markdown headers (##). No intro/outro filler. Bold the section labels.

User Prompt

The following are summaries of a scientific paper's individual sections:

Section {1} Summary:
{summary_1}

Section {2} Summary:
{summary_2}

[...]

Section {x} Summary:
{summary_x}

Generate a comprehensive bulleted summary for these sections now. Ensure that no critical methodological detail or key finding is omitted, as this summary will serve as the sole basis for a formal abstract.

Figure 6: The prompt template used to query `Qwen3-4B-Instruct-2507 4-bit` for papers that exceed the context window and are first split into chunks. `{summary_x}` is replaced with the summary of chunk x. The `[...]` notation is a schematic abbreviation; the actual prompt contains one numbered section block per chunk.

System Prompt

You are a senior scientific editor.
Your goal: Synthesize provided informations into a high–density, bulleted summary that contains every detail necessary to write a formal abstract.

Constraints:
1. Length: 400–500 words. Focus on maximum information density.
2. Structure: Use exactly four bulleted sections: CONTEXT (problem/objective), METHODOLOGY (design/sample/tools), RESULTS (specific findings/data), and CONCLUSION (significance/implications).
3. Requirements: DO NOT omit specific metrics, sample sizes, or p–values. Ensure the logical thread from hypothesis to implication is intact.
4. Use bullet points only.
5. Format: PLAIN TEXT ONLY. No Markdown headers (##). No intro/outro filler. Bold the section labels.

User Prompt

The following is the paper you are required to summarize:

PAPER START:

{content}

Generate a comprehensive bulleted summary for these sections now. Ensure that no critical methodological detail or key finding is omitted, as this summary will serve as the sole basis for a formal abstract.

Figure 7: The prompt template used to query `Qwen3-4B-Instruct-2507 4-bit` for papers that fit within the model's context window. `{content}` is replaced with the plain text of the paper.

System Prompt

You are an expert academic writer and stylist. Your task is to draft a formal academic abstract by synthesizing information from a summary while strictly adopting a clear, professional, and objective academic tone.

Guidelines:
1. Composition: Construct a cohesive abstract based on the provided 'Summary'.
2. Style: Apply standard academic vocabulary, precise sentence structures, and logical rhetorical patterns typical of high–quality research papers.
3. Fact Integrity: Use only the information provided in the summary. Do not hallucinate data, results, or citations not present in the source text.
4. Professionalism: Ensure the output follows standard academic conventions for abstracts (objective, concise, and impactful).
5. Constraints: Output only the final abstract. Do not include anything else like headers, remarks, meta–comments, or explanations.

User Prompt

Please compose a formal academic abstract based on the following summary.

<summary>
{summary}
</summary>

Output only the final abstract without a header:

Figure 8: The prompt template used for zero-shot, manual steering, learned steering, and hypernetwork setup. `{summary}` is replaced with the oueline of the paper.

System Prompt

You are an expert academic writer and stylist. Your task is to draft a formal academic abstract by synthesizing information from a summary while strictly adopting the linguistic style, tone, and structural preferences of example abstracts from a specific Author.

Guidelines:
1. Composition: Construct a cohesive abstract based on the provided 'Summary'.
2. Analyze the 'Author Examples' for sentence structure, vocabulary complexity, tone, and flow.
3. Style Mimicry: Apply the specific vocabulary, sentence structure, and rhetorical patterns associated with the given author examples.
4. Fact Integrity: Use only the information provided in the summary. Do not hallucinate data, results, or citations not present in the source text.
5. Professionalism: Ensure the output follows standard academic conventions for abstracts (objective, concise, and impactful).
6. Constraints: Output only the final abstract. Do not include introductory remarks, meta–comments, or explanations.

User Prompt

Here are the examples of the author's writing style:

<style_examples>
Example 1:
{example_1}

Example 2:
{example_2}

Example 3:
{example_3}
</style_examples>

Based on the style of the examples above, compose a formal academic abstract based on the following summary, written in the characteristic style of the specified examples.

<summary>
{summary}
</summary>

Output only the rewritten abstract:

Figure 9: The prompt template used for the few-shot and LoRA. `{summary}` is replaced with the outline of the paper. `{example_1}`, `{example_2}` and `{example_3}` are replaced with the example abstracts of the author.

(24 GB VRAM) with an Intel Xeon Silver 4110 CPU and 376 GiB of RAM. Qwen3-4B-Instruct-2507 is loaded in full precision for generation; the judge is loaded in 4-bit. The longest prompt in the dataset is 1,851 tokens for outline-only inputs and 2,740 tokens with three in-context examples, both well within the context limit, so no input truncation occurs.

Author Classifier The task is multiclass classification from an abstract to its first author. We search over four encoders (`bert-base-uncased`, `bert-base-multilingual-cased`, `distilbert-base-uncased`, `distilbert-base-multilingual-cased`) and three modes: a classification head, k-NN on frozen embeddings, and k-NN on fine-tuned embeddings with the head removed. For the head mode we search six learning rates (1×10^{-6} to 5×10^{-5}), selecting by validation loss. For the retrieval modes we search two distance functions (L2, cosine) and $k \in \{1, 3, 5, 7, 9\}$, selecting by validation accuracy. Training runs for up to 100 epochs with batch size 16 and AdamW, evaluating each epoch, with early stopping after five epochs without improvement and restoration of the best checkpoint. All abstracts fit within the 512-token limit, so no truncation is applied. The selected configuration is `distilbert-base-uncased`, fine-tuned k-NN, $k = 3$, cosine distance, learning rate 2×10^{-5}. The k-NN database holds all training abstracts; for test-set evaluation the validation abstracts are added, and for the unseen-author set it holds the three reference abstracts per author.

LoRA Baseline Adapters are applied to `q_proj` and `v_proj` with $r = 8$ and $\alpha = 16$. We use AdamW with a learning rate of 2×10^{-4}, 5 warmup steps, weight decay 0.001, and NEFTune (Jain et al., 2024) with noise $\alpha = 5.0$. The effective batch size is 8 (per-device 2, 4 gradient accumulation steps), and sequences are padded or truncated to 4,096 tokens. Cross-entropy is computed on the abstract only, with the prompt masked. Training runs for at most 50 epochs, evaluating every 25 steps, with early stopping after 25 evaluations without validation-loss improvement and restoration of the best checkpoint.

Style Control Networks The author representation is produced by a frozen STAR encoder; a linear projection maps its 1,024-dimensional output to 128 dimensions before averaging over the three references. The layer encoder maps the layer index to a 36-dimensional embedding followed by layer normalization; the module encoder (hypernetwork only) maps $\{\texttt{q\_proj}, \texttt{v\_proj}\}$ to a 32-dimensional embedding. Input dimensions are therefore $128 + 36 = 164$ for the steering network and $128 + 36 + 32 = 196$ for the hypernetwork. The backbone is a stack of four MLP blocks with a 512-dimensional up-projection, SiLU activations, and 5% dropout: the first block omits layer normalization, the second and third are residual blocks with pre-normalization, and the fourth normalizes without a residual connection. Backbone weights use Kaiming normal initialization with zero biases. The steering network has a single head projecting $512 \to 2{,}560$, zero-initialized in weights and bias. The hypernetwork has four heads emitting flattened adapters at $r = 8$, with shapes following Equation 4, i.e. $\mathbf{A} \in \mathbb{R}^{r\times d_{\text{in}}}$ and $\mathbf{B} \in \mathbb{R}^{d_{\text{out}}\times r}$: $\texttt{q}_A$ (8×2,560), $\texttt{q}_B$ (4,096×8), $\texttt{v}_A$ (8×2,560), and $\texttt{v}_B$ (1,024×8). The A-heads have zero weights and biases drawn uniformly from $[-1/\sqrt{d_{\text{in}}}, 1/\sqrt{d_{\text{in}}}]$; the B-heads are zero in both, matching standard LoRA initialization. Training follows the LoRA baseline's configuration, with mixed precision (fp16) and gradient checkpointing added to reduce memory.

Listings 1–2 show the full architecture of the shared backbone used by both the activation vector prediction network and the hypernetwork model. The input module (Listing 1) encodes the author style embedding, the target layer index, and, for the hypernetwork, the target module. The backbone (Listing 3) consists of four sequential MLP blocks. The output heads (Listing 2) differ between the two models: the hypernetwork uses the four LoRA matrix heads, while the activation vector prediction network uses only the single steering vector head.

```
Input(
  (star_proj): Sequential(
    (0): Linear(in_features=1024,
    out_features=128, bias=True)
    (1): LayerNorm((128,), eps=1e-05,
    elementwise_affine=True)
  )
  (layer_depth_encoder): Sequential(
    (0): Embedding({num_of_layers}, 36)
    (1): LayerNorm((36,), eps=1e-05,
    elementwise_affine=True)
  )
  (module_emb): Sequential(
    (0): Embedding(2, 32)
    (1): LayerNorm((32,), eps=1e-05,
    elementwise_affine=True)
  )
)
```

Listing 1: Input to the style control networks. For the activation vector prediction network, the module embedding is removed. All inputs are concatenated to form the final input to the network. `num_of_layers` is replaced with the number of layers that the model is controlling.

```
Output(
  (final_proj_q_A): Linear(in_features
    =512, out_features=20480, bias=True)
  (final_proj_q_B): Linear(in_features
    =512, out_features=32768, bias=True)
  (final_proj_v_A): Linear(in_features
    =512, out_features=20480, bias=True)
  (final_proj_v_B): Linear(in_features
    =512, out_features=8192, bias=True)

  (final_steering): Linear(in_features
    =512, out_features=2560, bias=True)
)
```

Listing 2: Output of the style control networks. Either the first four output heads are used for the LoRA hypernetwork or the `final_steering` output head is used for the activation vector prediction network. Each head emits a flattened matrix; the reshaped targets follow Equation 4.

```
ModuleDict(
  (mlp1): MLPResidualBlock(
      (mlp): Sequential(
        (0): Linear(in_features={
    hidden_dim}, out_features=512, bias=
    True)
        (1): SiLU()
        (2): Dropout(p=0.05, inplace=
    False)
        (3): Linear(in_features=512,
    out_features={hidden_dim}, bias=True
    )
        (4): SiLU()
        (5): Dropout(p=0.05, inplace=
    False)
      )
  )
  (mlp2): MLPResidualBlock(
    (mlp): Sequential(
      (0): LayerNorm(({hidden_dim},),
    eps=1e-05, elementwise_affine=True)
      (1): Linear(in_features={
    hidden_dim}, out_features=512, bias=
    True)
      (2): SiLU()
      (3): Dropout(p=0.05, inplace=False
    )
      (4): Linear(in_features=512,
    out_features={hidden_dim}, bias=True
    )
      (5): SiLU()
      (6): Dropout(p=0.05, inplace=False
    )
    )
  )
  (mlp3): MLPResidualBlock(
    (mlp): Sequential(
      (0): LayerNorm(({hidden_dim},),
    eps=1e-05, elementwise_affine=True)
      (1): Linear(in_features={
    hidden_dim}, out_features=512, bias=
    True)
      (2): SiLU()
      (3): Dropout(p=0.05, inplace=False
    )
      (4): Linear(in_features=512,
    out_features={hidden_dim}, bias=True
    )
      (5): SiLU()
      (6): Dropout(p=0.05, inplace=False
    )
    )
  )
  (mlp4): Sequential(
    (0): LayerNorm(({hidden_dim},), eps
    =1e-05, elementwise_affine=True)
    (1): Linear(in_features={hidden_dim
    }, out_features=512, bias=True)
    (2): SiLU()
    (3): Dropout(p=0.05, inplace=False)
    (4): Linear(in_features=512,
    out_features=512, bias=True)
    (5): SiLU()
  )
)
```

Listing 3: Backbone of the style control networks. The input is passed through the network in the order of the listing. `hidden_dim` is 196 for the LoRA hypernetwork and 164 for the activation prediction network.

QUERYDIAGRAMEVAL: Evaluating and Benchmarking LLM Diagram Generation under Noisy Conditions

Ted Zhang[1], Souvik Das[1], Kai Ni [1], Aditya Agrawal [1], Xin Guo [1], Denis Kochedykov [1]
[1]J.P.Morgan Chase & Co.
ce.zhang@jpmchase.com

Abstract

Chat assistants increasingly ask large language models to turn free-form user prose into structured, executable diagrams. These requests are noisy user-generated text: informal, hedged, jargon-mixed, incomplete, and often describing the target only indirectly—in workflow language rather than diagram vocabulary. Evaluation is hard because many distinct diagrams satisfy the same noisy request, making reference-based metrics too rigid where no ground truth exists, while reference-free LLM judges give weak, hard-to-reproduce diagnostics. We introduce QUERYDIAGRAMEVAL (QDE), a reference-free protocol scoring diagram generation directly against raw, noisy queries via two levels: deterministic compilation gates (syntax, rendering, diagram-type correctness), then a semantic-fidelity layer where a multimodal auditor is grounded in type-specific structural facts parsed from the code. On 255 annotated predictions, QDE aligns strongly with holistic human ratings ($\rho = 0.715$), outperforming an ungrounded LLM judge. We also release MERMAIDBENCH, a source-grounded benchmark of 1,150 query–gold pairs over 11 diagram types, four query-difficulty tiers, and a multi-turn track, with the gold held fixed to isolate robustness to input-side noise. Furthermore, we conduct experiments on eight frontier-model and harnesses to show that the our agentic framework reaches near-gold quality while failures concentrate in topology preservation, label grounding, and type-specific syntax.

1 Introduction

Large language models (LLMs) are increasingly asked to turn *user-generated natural-language requests* into structured, executable artifacts (Naveed et al., 2024). Diagram generation is a particularly demanding case: in chat assistants such as GitHub Copilot, Claude Code, and ChatGPT, users routinely ask models to turn free-form prose into flowcharts, sequence diagrams, ER diagrams, Gantt charts, and other visual specifications (Zala et al., 2024). In production these requests are *noisy user text*: informal, hedged, jargon-mixed, incomplete, and often describing the target only *indirectly*—in workflow language rather than diagram vocabulary. The gap between what the user typed and what the diagram must depict is precisely the noisy-text challenge, but the downstream artifact is now a compilable typed visual program.

Evaluation methods inherited from code-generation benchmarks such as HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), and SWE-bench (Jimenez et al., 2024) do not transfer: diagrams compile into visual artifacts rather than executable behaviour, and many structurally distinct diagrams may all be valid answers to a single noisy query, making exact-match and pixel-similarity metrics against a gold reference brittle (Liang and You, 2025). Deployment is also inherently *reference-free*—only the user's noisy query is available at inference—so reference-based metrics are unusable for online evaluation, A/B testing, or guardrails on user traffic. Silent semantic failures (a rendered flowchart with a reversed arrow or a missing service) are far more costly than syntax errors, yet "does it render?" CI checks miss them and human review does not scale.

We study these challenges in Mermaid [1], a widely deployed text-to-diagram language whose typed structure (flowchart nodes/edges, sequence participants/messages, ER entities/relations, etc.) enables automatic checking while still requiring the evaluator to judge intent expressed in *unconstrained, potentially noisy* natural language.

We introduce a reference-free evaluation protocol, QUERYDIAGRAMEVAL (QDE), and a Mermaid benchmark, MERMAIDBENCH, that jointly stress-test diagram generation *under noisy user text*. QDE first applies deterministic compilation

[1]https://mermaid.js.org/

Proceedings of the 11th Workshop on Natural User-generated Text, pages 244–265
October 28, 2026 ©2026 Association for Computational Linguistics

gates (syntax, render, type), then extracts type-specific structural facts from the code and passes them—together with the raw user query and rendered image—to a multimodal auditor that performs a fact-anchored requirement audit; k=5 self-consistent samples are averaged. MERMAIDBENCH contains 1,150 tasks derived from 230 expert-verified real-world diagrams. Each source is fanned out into four single-turn *query tiers*—clean explicit spec (T1), adversarial noise and hedging (T2), workflow phrasing with no diagram terms (T3), indirect scenario (T4)—plus a multi-turn refinement track, with the gold diagram held fixed across tiers so scores isolate robustness to *input-side noise* across all 11 stable Mermaid types.

Across eight frontier configurations, QDE correlates strongly with human holistic ratings (ρ=0.715). Query indirection is the dominant difficulty axis (-1.8 T2→T3, -2.1 T3→T4), confirming that current LLMs remain brittle when user text drifts from clean specs, and residual failures are semantic—missing/hallucinated topology, label mismatches, type-specific slips—rather than syntactic.

Contributions.

1. QUERYDIAGRAMEVAL, a reference-free two-level protocol for scoring LLM diagram generation against *raw user queries* (ρ=0.715 with human ratings).
2. MERMAIDBENCH, a source-grounded benchmark of 1,150 tasks over 11 Mermaid types, four *noise-controlled* query tiers, and a multi-turn refinement track.
3. MERMAIDAGENT, a ReAct-based tool-augmented reference architecture evaluated across the full noise spectrum.

2 Related Work

Diagram evaluation. Prior diagram-quality metrics fall into three families. *Pixel-level similarity*, used by AutomaTikZ (Belouadi et al., 2024) (CLIPScore) and DiagrammerGPT (Zala et al., 2024) (raster outputs), measures visual closeness to a reference image, ignoring whether the underlying code is structurally correct or grammar-valid. *Reference-based graph metrics* address this by parsing diagrams into graphs: DiagramEval (Liang and You, 2025) computes node- and path-alignment F1 between predicted and gold graphs, while DiagramAgent (Wei et al., 2024) reports task-specific checks against gold templates. *LLM-as-judge* approaches such as MT-Bench (Zheng et al., 2023) and the Mermaid-specific MermaidSeqBench (Shbita et al., 2025) score outputs holistically but inherit verbosity, position, and self-enhancement biases. All three families either require a gold reference, degenerate on list-oriented types (Gantt, timeline, journey, quadrant), or expose the final score to unconstrained judge bias. QUERYDIAGRAMEVAL (§3) closes this gap by anchoring a multimodal LLM judge with type-specific structural facts deterministically parsed from the candidate code, enforcing Level 1 compilation gates, and aggregating k=5 self-consistent samples—producing a *reference-free* score that correlates with human ratings at ρ=0.715.

Benchmarks for structured code generation. Program-generation benchmarks such as HumanEval (Chen et al., 2021), MBPP (Austin et al., 2021), DS-1000 (Lai et al., 2023), and SWE-bench (Jimenez et al., 2024) grade programs through execution, providing no signal for visual or structural artefacts. On the diagram side, ChartQA (Masry et al., 2022) and PlotBench (Zhao et al., 2025) target charts rather than typed diagrams; AutomaTikZ covers a single TikZ format with no difficulty control; MMCode (Li et al., 2024) targets general multimodal programming; DiagramAgent's DiagramGenBench is synthetically constructed across eight categories without source-grounded golds; MermaidSeqBench covers only sequence diagrams (one of 11 Mermaid types) with 132 samples. Synthetic construction approaches like Self-Instruct (Wang et al., 2023) and Evol-Instruct (Xu et al., 2024) risk circular bias when benchmark artefacts are pure LLM self-play. MERMAIDBENCH (§4) fills this gap with 340 test tasks (1,150 total) spanning 11 Mermaid types, four query tiers, and a multi-turn track, with gold diagrams distilled from real-world images and human-audited. None of these prior protocols are directly deployable in the reference-free regime that production diagram assistants actually face; MERMAIDBENCH and QDE combine broad typed Mermaid coverage with source-grounded construction and reference-free scoring, making them suitable both as a research benchmark and as an in-production quality gate.

3 QueryDiagramEval

Diagram code admits multiple valid realizations for any given query, making reference-based

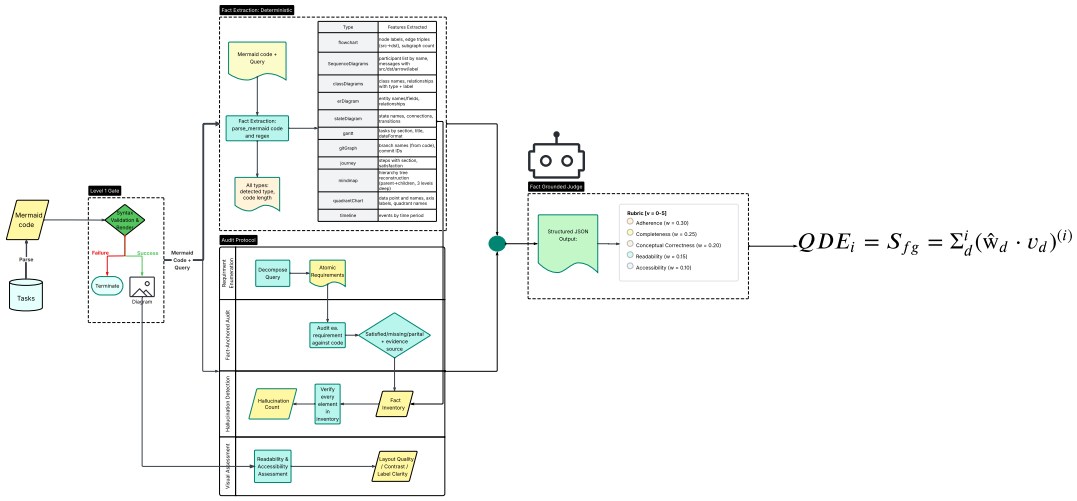


Figure 1: The two-level QUERYDIAGRAMEVAL (QDE) protocol. Level 1 runs a deterministic parser and renderer plus a type check whose outcomes act as hard gates. Level 2 extracts type-specific structural facts from the candidate code and provides them, together with the query and rendered image, to a multimodal auditor that produces a fact-grounded rubric score. QDE is reference-free: only the query and predicted code are required.

exact-match metrics unreliable. Relying solely on an LLM-as-judge, in contrast, conflates semantic assessment with structural and compilation correctness, sacrifices reproducibility, and inherits known judge biases (verbosity preference, self-enhancement; Zheng et al., 2023). We therefore propose QUERYDIAGRAMEVAL (QDE), a *reference-free, two-level* protocol (Figure 1) that combines **compilation verification** (§3.1)—deterministic checks for syntax validity, render success, and type correctness, enforced as hard gates—with **fact-grounded semantic assessment** (S_{fg}, §3.2)—a multimodal LLM auditor anchored by deterministic structural facts extracted from the candidate code, aggregated by $k{=}5$ self-consistent samples.

Given a user query q and predicted Mermaid code $\hat{y}$, the QDE score is

$$\text{QDE} = S_{\text{fg}} \quad \text{subject to Level 1 hard gates.} \quad (1)$$

This design ensures that (i) structurally broken code cannot achieve high scores regardless of auditor output; (ii) the auditor operates on verifiable structural evidence rather than surface impressions; and (iii) Level 1 metrics (render rate, type accuracy) are fully deterministic and independently reportable.

3.1 Level 1: Compilation Verification

The mermaid code block is extracted from the response and parsed by a type-aware AST parser supporting all 11 stable Mermaid types. The parser identifies the declared type and extracts typed structural elements (nodes, edges, subgraphs, participants, tasks, commits, entities, etc.). Valid code is rendered to SVG and PNG.

Three binary signals—*parse pass rate*, *render success rate*, and *type accuracy*—are recorded as independent metrics and enforced as hard gates. If any of the binary signals fails the candidate mermaid code is rejected therefore preventing QDE from generating a unreliable score, especially if the mermaid code contains syntax errors that make rendering impossible.

3.2 Level 2: Fact-Grounded Semantic Assessment

Type-specific fact extraction. Pure LLM-as-judge evaluation operates on surface impressions of the code and image, without verified structural information; the auditor may hallucinate elements, miscount entities, or overlook structural deficiencies. QDE grounds the auditor with deterministic structural facts extracted from the candidate code. A type-aware parser produces a fact summary tailored to the diagram type: flowchart nodes/edges/subgraphs, sequenceDiagram participants and messages with labels, classDiagram classes and typed relationships, erDiagram entities and relationships, stateDiagram states/transitions, gantt tasks/sections, gitGraph branches/commits, journey steps with satisfaction scores, mindmap hierarchy trees, quadrantChart data points and axes,

and timeline events (Appendix B, Table 9). These facts are deterministic and verifiable—they represent ground truth about what the code actually contains.

Fact-anchored requirement audit. A multimodal auditor receives four inputs: the user query q, the Mermaid source $\hat{y}$, the rendered PNG, and the structural fact summary. It performs a five-step chain-of-thought assessment: (i) decomposes q into atomic requirements; (ii) classifies each as satisfied, missing, or partial by cross-referencing the structural facts and its own semantic analysis, tagging each decision with its evidence source (`structural_fact`, `semantic`, or `image`); (iii) flags hallucinated elements; (iv) scores readability and accessibility from the rendered image; and (v) returns a JSON object with per-requirement statuses, error counts, and five 0–5 dimension scores. The complete system prompt, structural-fact template, and per-task user message appear in Appendix D.

Rubric aggregation and self-consistency. The five dimension scores $v_d \in \{0, \ldots, 5\}$ are aggregated with fixed weights $\hat{w} = (0.30, 0.25, 0.20, 0.15, 0.10)$ over (adherence, completeness, conceptual correctness, readability, accessibility) and scaled to $[0, 100]$ as shown in equation Equation 2 Adherence and completeness dominate because diagram-generation failures are usually semantic. To reduce stochastic judge variance we sample the audit k=5 times and take the mean:

$$S_{\text{fg}}^{(i)} = 20 \sum_{d} \hat{w}_d v_d^{(i)}. \tag{2}$$

On the validation pool (N=255), the mean inter-sample standard deviation is 2.4 points on 0–100 (median spread 4.0); only 14% of items exceed σ=5. The per-task $\bar{\sigma}_k$ is reported alongside model scores (Table 2) as a judge-stability diagnostic.

Why reference-free? Reference-based structural metrics (e.g., graph F1 against gold code) penalise valid alternative realizations—paraphrased labels, reordered nodes, equivalent subgraph decompositions. On our validation pool, a reference-based deterministic audit achieves only ρ=0.462 with human ratings, versus ρ=0.715 for the fact-grounded reference-free auditor. More practically, reference-free evaluation is a strict prerequisite for online deployment (CI/CD, reward models, quality gates), where gold references do not exist.

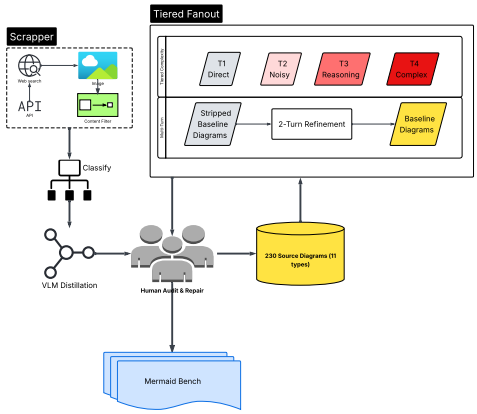


Figure 2: MERMAIDBENCH construction pipeline. Real-world diagrams are collected, distilled to Mermaid via a VLM using a describe-then-extract protocol, expert-audited and repaired, then fanned out into four query tiers plus a multi-turn track, quality-gated, and split into dev/test sets.

4 MermaidBench

MERMAIDBENCH is designed to be *source-grounded* (anchored to real-world diagrams, not LLM self-play), *type-diverse* (11 Mermaid types), *difficulty-controlled* (query-side only), and *contamination-resistant* (canary identifiers).

Source collection and VLM-assisted authoring. We source 323 candidate technical diagrams from academic papers, technical documentation, and image search. An LLM-first classifier retains true technical diagrams and predicts a canonical Mermaid type; cheap structural gates (aspect ratio, minimum resolution, text-region count) reject obvious non-diagrams. After quality filtering and type balancing, 230 sources enter the authoring pipeline. For each source, a VLM produces baseline Mermaid code via a *describe-then-extract* protocol: it first produces a detailed textual description of the diagram's components, relationships, and layout, then generates Mermaid targeting the predicted type conditioned on both image and description. Distillation runs in a 3-retry loop with rendering-based validation and *fidelity-guided* feedback: type-specific structural checks (e.g., node/edge counts for flowcharts, participant/message counts for sequence diagrams) supply targeted feedback on failures.

Expert audit and repair. All 230 baselines undergo expert audit for type fidelity, element coverage, structural accuracy, hallucinations, and label faithfulness; annotators repair every identified error. The VLM draft functions as a productivity accelerator rather than the ground truth: 100% of released baselines are human-verified.

Tier fan-out and multi-turn track. Every verified baseline is converted into four single-turn query tiers plus one multi-turn task. **T1 (Direct)** names diagram elements explicitly using Mermaid terminology; **T2 (Adversarial)** introduces noise, distractors, and hedging around the same requirements; **T3 (Reasoning)** uses workflow or business phrasing with no Mermaid jargon; **T4 (Complex)** describes an indirect scenario requiring inference of diagram structure. The gold code is held fixed across tiers, isolating query-comprehension difficulty from gold-side variation. Multi-turn tasks start from a stripped-down baseline (edges and styling removed); the user acknowledges the simplified diagram and then requests specific improvements toward the full baseline, isolating iterative-refinement capability. A representative tier progression for a single source diagram appears in Appendix A, Figure 4.

Contamination resistance. We inject unique canary identifiers into gold node IDs during baseline construction: each baseline receives three mutations following `XQV{NNNN}_{shortid}` (globally unique per run), and a quality gate verifies that the mutated code still renders. Because the tier queries are authored from the canary-mutated gold, a canary token can legitimately surface in the query itself (and, for multi-turn tasks, in the simplified diagram shown to the model), so a model can reproduce a canary simply by *copying* it from its own input. We therefore do *not* treat mere canary appearance as memorization. Contamination is flagged only by a *hidden-canary* match—a canary that appears in a generation but in *none* of the model-visible inputs (query plus any conversation context). The public dev split has canaries reverted to their original identifiers before release, while the test split retains them for this probe. Across all evaluated systems we observe *zero* hidden-canary matches, i.e., no detectable memorization of test golds; full procedure and counts are in Appendix A.1.

Split	Base	Single-turn	Multi-turn	Total
Dev	162	648	162	810
Test	68	272	68	340
All	230	920	230	1,150

Table 1: Dataset scale. Full per-type composition appears in Appendix A, Table 7.

Quality gates and splitting. Baselines must pass three automated gates before release: (i) parses and renders via the pinned Mermaid.js runtime; (ii) canary survival; and (iii) readability thresholds—label overlap < 0.08, whitespace ratio $\in [0.30, 0.85]$, aspect ratio $\in [0.3, 3.0]$, plus type-specific extractability. The fan-out produces $230 \times 4 = 920$ single-turn tasks and 230 multi-turn tasks (1,150 total). A stratified 70/30 dev/test split (810/340) is computed with a fixed seed and an anti-leakage constraint that prevents variants of the same source from crossing splits.

5 MermaidAgent

MERMAIDAGENT is a tool-augmented language agent that generates Mermaid diagrams from natural-language queries under the Reasoning-and-Acting paradigm (Yao et al., 2023). It serves both as a competitive system on MERMAIDBENCH (§6) and as a reference architecture for grounding LLM generation in deterministic Mermaid tooling. The agent separates *planning* (what to draw), *generation* (how to draw it), and *evaluation* (whether the draft is correct). Diagram generation from natural language poses three compounding challenges: (i) Mermaid's strict, type-specific syntax means a single malformed token blocks rendering; (ii) semantic correctness cannot be verified by a rule-based check; and (iii) different diagram varieties impose fundamentally different structural constraints. Each LLM-driven stage is therefore grounded by deterministic tools that supply authoritative syntax and validate output.

Planner. Two sequential LLM calls, each constrained to a structured output schema, produce first an *intent analysis* (closed-vocabulary diagram type, estimated complexity, subject domain, ranked key requirements, detected ambiguities, style constraints) and then a *diagram plan* (structural outline, key nodes/entities, edges/relationships, styling approach, and a justification of complexity). The closed type vocabulary bounds the search space; the plan is injected into the generation prompt as a soft constraint, preserving generative flexibility

while maintaining intent alignment.

Tool registry. Four tools support the agent. Two are knowledge tools consulted once per request: $\mathcal{T}_1$ (documentation) returns a 700–1,000-character authoritative syntax summary for the target type; $\mathcal{T}_2$ (design principles) injects expert guidelines along four axes—general clarity/consistency, type-specific rules, styling, and complexity-level element-count targets. Two are action tools invoked inside the ReAct loop: $\mathcal{T}_3$ (repair) provides two-stage syntax validation, structure analysis, and rule-based fix suggestions; $\mathcal{T}_4$ (critic) performs deterministic multi-dimensional evaluation and returns syntax validity, type correctness, a requirement-coverage score s_{cov}, structural-completeness flags, and a syntax-weighted alignment score $s_{\text{align}} \in [0, 1]$. Full tool schemas, prompts, the coverage/alignment formulas, and the approval predicate (syntax_valid $\wedge$ type_match $\wedge$ $|\text{design\_issues}| \leq 1 \wedge s_{\text{cov}} \geq 0.7$) are reproduced in Appendix E (Figure 5).

ReAct loop. The core loop runs up to $N_{\max} = 5$ Thought–Act–Observe iterations. Iteration 1 composes a generation prompt from the query, intent, plan, syntax reference, and design principles, and applies structural validation immediately; on failure a focused syntax-only sub-routine constrained to minimal edits fires before critique, preventing a trivially invalid draft from consuming a full iteration. Iterations 2–$N_{\max}$ consume critic feedback, emit a structured repair plan, and apply a full repair whose prompt includes the plan and the first 1,000 characters of the syntax reference as a grounding guard. The loop exits early when the critic returns approved $=$ true with $s_{\text{align}} \geq 0.5$; otherwise the last candidate is returned with a warning. Every step is trace-logged; a 2–3-sentence caption is produced at termination.

6 Experiments

6.1 Setup

We report two studies over disjoint data partitions. A *metric-validation study* (§6.2) tests QDE–human alignment on a quality-stratified pool of 255 predictions with three human ratings each (inter-annotator Krippendorff's α=0.797 on holistic ratings; Krippendorff, 2011; full annotator, training, blinding, rubric, adjudication, and per-dimension agreement details in Appendix C). A *model-comparison study* (§6.3) evaluates eight configurations on the 340-task MERMAIDBENCH test set (272 single-turn + 68 multi-turn), scoring every model output against its authored query. Direct configurations use standard prompting; reasoning configurations enable extended thinking; agentic configurations use tool-augmented iterative refinement—the Mermaid-specific MERMAIDAGENT (§5) and a Mermaid port of DiagramAgent (Wei et al., 2024) sharing the same pinned Mermaid backend.

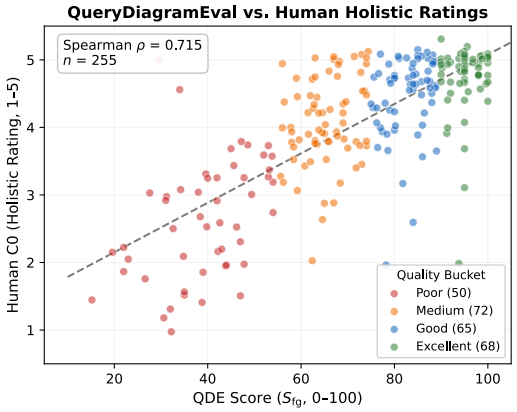


Figure 3: QDE vs. human holistic ratings (C0) on the validation pool (n=255). Points are coloured by quality-stratification bucket. Spearman $\rho = 0.715$.

6.2 Metric Validation

The validation pool combines real model outputs (∼60%) with controlled gold-code degradations (∼40%). We degradated the gold code with label scrambling, partial omits, edge deletion and node hallucination. The original benchmark query is always preserved. Predictions are then bucketed by automatic QDE score into four quality bands before annotation so that no single quality band dominates the correlation estimate. Because this stratification deliberately spans the metric's range, we also report correlation on the naturally-distributed real-output subset alone (ρ=0.444, n=134; attenuated by ceiling concentration), which remains positive and significant in Section 7 the Discussion.

QDE achieves the strongest alignment with human ratings: the fact-grounded S_{fg} reaches ρ=0.715 (Table 3), compared with ρ=0.668 for an ungrounded chain-of-thought judge (+0.047; improvements on 9 of 11 types) and ρ=0.462 for a reference-based structural score (GRAPHF1, a deterministic node/edge-set F1 against the gold code adapted from the graph-based diagram metric of Liang and You, 2025; full specification, including label normalization, node alignment, edge-direction handling, and per-type behavior on non-

Model	QDE by Tier T1	T2	T3	T4	QDE Overall	$\bar{\sigma}_k$	MT QDE	Render %
GPT-5.4	68.9	76.8	74.6	70.8	75.1	1.3	84.2	87.1
GPT-5.4 (reasoning)	81.2	75.0	76.8	65.6	76.7	1.1	84.8	85.9
Claude Opus 4.6	82.6	83.7	**87.0**	83.0	85.4	1.3	**90.9**	97.1
Claude Opus 4.6 (reasoning)	85.9	85.7	85.9	**84.2**	85.4	1.6	79.7	**98.0**
DiagramAgent w/ GPT-5.4	79.6	81.8	74.7	73.8	77.5	1.6	77.7	94.4
DiagramAgent w/ Claude Opus 4.6	80.1	83.2	77.7	76.5	81.3	1.3	87.7	95.2
MermaidAgent w/ GPT-5.4	77.0	70.8	69.0	77.6	73.6	1.6	86.7	88.7
MermaidAgent w/ Claude Opus 4.6	**88.4**	**87.9**	85.0	82.8	**86.7**	1.3	89.4	96.5
Gold baseline	88.9	90.1	87.9	87.8	89.5	1.5	95.6	100.0

Table 2: Main results on the MERMAIDBENCH test set (n=340). QDE is on 0–100. MT = multi-turn. $\bar{\sigma}_k$ = mean per-task standard deviation across k=5 judge samples. Type accuracy (not shown): $\geq 86.8\%$ for all models. Per-type breakdowns and tier deltas appear in Appendix F.

Metric variant	Level	Spearman ρ
Graph F1	Ref.	0.462
S_{cot} (vanilla judge, no facts)	Ablation	0.668
S_{fg} (fact-grounded, k=5)	L2 (Semantic)	**0.715**

Table 3: Metric correlation with human holistic ratings (C0) on the validation pool (N=255). Level 1 binary metrics are hard gates rather than continuous predictors. 95% CI for S_{fg}: $[0.640, 0.778]$.

graph diagrams, in Appendix F.3). The largest gains occur on structurally rich types such as class diagrams (+0.160), ER diagrams (+0.140), and journey diagrams (+0.093), where deterministic facts help the auditor count entities and relations reliably. Decomposing by rating dimension, semantic dimensions align most strongly with humans—adherence (ρ=0.699), conceptual correctness (0.654), completeness (0.582)—while visual dimensions are weaker but still positive (readability 0.316, accessibility 0.256). Per-bucket and dimension-wise tables appear in Appendix F.2.

6.3 Main Results

Table 2 reports the primary comparison. All 340 gold codes pass Level 1 gates and receive mean QDE = 89.5 (median 92.0; per-type range 79.4–93.8), confirming that the metric assigns high scores to known-correct code while retaining sensitivity. Because QDE is reference-free, a model can occasionally exceed the gold baseline on individual slices if it produces cleaner, more judge-amenable syntax than the human author; the gold row should therefore be read as a strong human reference point rather than a hard upper bound.

Three patterns stand out. **(1) Agentic refinement helps when paired with the strongest base model.** MERMAIDAGENT with Claude Opus 4.6 achieves the highest overall QDE (86.7) while maintaining 96.5% render success, whereas the same scaffold on GPT-5.4 underperforms direct prompting (73.6 vs. 75.1)—suggesting that iterative repair loops magnify base-model quality rather than substituting for it. All three Claude configurations achieve $\geq 96\%$ render success. **(2) Reasoning has mixed effects.** GPT-5.4 reasoning slightly exceeds direct (76.7 vs. 75.1), primarily from a T1 boost (+12.3), but degrades on T4 (−5.2). For Claude Opus 4.6, reasoning matches direct on aggregate (85.4 vs. 85.4) with 98.0% render success, indicating that extended thinking does not degrade Mermaid generation but provides little marginal benefit at the frontier. **(3) Multi-turn dialogue often helps direct models.** GPT-5.4 direct improves from 72.8 single-turn to 84.2 multi-turn (+11.4), and MERMAIDAGENT/GPT-5.4 gains +13.1, consistent with multi-turn decomposing complex requirements into smaller edits. The DiagramAgent scaffold and Claude Opus 4.6 reasoning are exceptions (+0.2 and −5.7 respectively), highlighting that the multi-turn advantage is configuration-dependent.

Query indirection remains the primary difficulty axis. On average across models, QDE drops −1.8 from T2 to T3 and −2.1 from T3 to T4 as queries shift from explicit to indirect phrasing, while T1→T2 is essentially flat (+0.2)—adversarial noise alone is comparatively benign. Full tier deltas and per-type scores are in Appendix F.4 and F.1.

7 Discussion

Error taxonomy. Human error tags from repairing gold diagrams and rating model outputs (Appendix F.6) show that dominant failure modes are structural rather than syntactic: missing edges (22.6%), missing nodes (19.0%), layout/readability

(15.1%), label mismatches (14.8%), hallucinated nodes (14.0%), over-specification (13.7%), and factual errors (12.0%). Syntax and type errors are comparatively rare among frontier models but remain important because they render outputs unusable. This supports QDE's design: deterministic Level 1 gates catch hard compilation failures, while fact-grounded semantic auditing exposes topology and label errors that "render success" alone cannot see.

Input-side verbosity leniency. Because every tier of a given baseline shares the same gold code, gold-diagram QDE scores across tiers isolate a pure evaluator effect. We observe an *input-side* verbosity leniency: verbose T2/T4 prompts sometimes receive higher S_{fg} than concise T3 paraphrases for the same gold, apparently because the auditor resolves ambiguity in favour of the candidate when requirements are diluted with distractors. In a matched comparison across 68 baselines, T4 receives higher QDE than T2 in 60% of cases (ΔQDE$= +8.9$), concentrated in completeness, readability, and accessibility dimensions. This complements known output-side verbosity bias in LLM judges (Zheng et al., 2023) and motivates future query-length normalization; absolute QDE scores should be compared within-tier.

Self-enhancement bias. Because Claude Opus 4.6 is both the default judge and a candidate generator, self-preference is a natural concern. Indirect evidence argues against a severe effect: (i) the human-authored gold baseline outscores every LLM-generated system (89.5); (ii) Claude direct and Claude reasoning receive identical aggregate QDE (85.4 vs. 85.4); (iii) on the shared DIAGRAMAGENT scaffold the Claude variant exceeds the GPT-5.4 variant by only 3.8 points, consistent with their respective render-success rates; and (iv) among three fact-grounded judges evaluated against human holistic ratings, Claude attains the highest correlation (ρ=0.715 vs. GPT-5.2 ρ=0.595 and GPT-4.1 ρ=0.568). Direct evidence: re-scoring a stratified random sample of 175 (model, task) pairs with GPT-4.1 as an alternative fact-grounded judge yields model-level Kendall τ=0.905 (Spearman ρ=0.964) with our main results; the top four systems retain their exact ranking (Appendix F.5). GPT-4.1 is systematically more lenient by 6–11 points, but the relative ordering that drives our conclusions is judge-invariant.

Implications. Because QDE is reference-free at inference time, it is directly deployable as (a) an offline benchmark metric, (b) an online CI/CD quality gate for LLM diagram assistants, and (c) a reward signal for fine-tuning diagram agents on natural queries where no gold exists. The Level 1 gates provide interpretable diagnostics—parse, render, and type accuracy—that decouple structural quality from semantic quality when tracking model progress.

Evaluation Datasets and Validation Pool The main results and the metric-validation results use different partitions and should not be conflated (Table 4). The validation pool tests QDE–human alignment across a broad quality range, whereas the held-out test set is used to compare model configurations on natural benchmark tasks.

	Validation pool	**Test set**
Purpose	Metric calibration	Model comparison
Section	§6.2	§6.3
Size (N)	255	340
Split	dev (annotation)	test
Prediction source	Real + degraded gold	Model outputs only
Query–gold pairing	Original (preserved)	Original (preserved)
Human ratings	✓ (3/task)	—

Table 4: The two evaluation datasets. They are disjoint and serve complementary roles.

Validation-pool composition. Table 5 lists the composition. Real predictions provide medium-to-high-quality outputs; controlled degradations create known failure modes. Perturbations are applied only to the candidate diagram code—the original query is preserved so that the pool tests whether QDE penalises diagrams that fail to satisfy a valid request without introducing query–gold mismatch as a confound.

	Source	**Approx. share**
Real predictions	GPT-4.1 (T1–T4 + MT)	∼60%
Degraded (gold-side)	`label_scramble`	∼19%
	`node_hallucination`	∼10%
	`edge_deletion`	∼9%
	`partial_omit`	∼6%
	`syntax_break`	∼3%

Table 5: Validation-pool composition. All items pair the original benchmark query with either a real model output or a perturbed gold diagram.

Stratified vs. naturally-distributed sampling. Because the pool is quality-stratified—and the synthetic degradations are constructed to span QDE's range—correlation on the full pool partly reflects the design's deliberate coverage of the metric's scale rather than an unbiased draw of naturally occurring outputs. To separate these effects, Table 6 recomputes $\rho(\text{C0}, \text{QDE})$ on three subsets: the full stratified pool, the *naturally-distributed* real model outputs only (no synthetic perturbation, not selected to span the metric), and the degraded-gold items only. As expected, the degraded subset carries most of the between-band signal (ρ=0.839) since perturbations create wide, easily separable quality gaps, while the natural real-output subset is lower (ρ=0.444). Crucially, the natural subset is strongly ceiling-concentrated (mean C0 = 4.46 vs. 4.08 pool-wide), so its reduced ρ is largely range restriction rather than metric failure—the same ceiling-compression effect documented in the appendix Table 17—and QDE still tracks human judgment positively and significantly on naturally occurring outputs (95% CI excludes zero). We report both numbers rather than the stratified figure alone. Two caveats remain: the natural subset covers only GPT-4.1 outputs (human ratings were not collected for the other Table 2 systems), and its ceiling concentration limits the achievable within-sample ρ; a larger natural-distribution annotation spanning all evaluated systems is left to future work.

Subset	n	mean C0	ρ(C0, QDE)
Full stratified pool	255	4.08	**0.715**
Real outputs only (natural)	134	4.46	0.444
Degraded gold only	121	3.66	0.839

Table 6: Correlation under stratified vs. naturally-distributed sampling. The natural (real-output) subset is not selected to span the metric and is ceiling-concentrated, which restricts its within-sample ρ; the correlation nonetheless remains positive and significant (95% CI $[0.296, 0.570]$).

8 Conclusion

We presented QUERYDIAGRAMEVAL, a reference-free two-level protocol for diagram-code evaluation that combines deterministic compilation gates with fact-grounded multimodal semantic assessment, and MERMAIDBENCH, a 1,150-task source-grounded benchmark spanning 11 diagram types, four query tiers, and a multi-turn track. Human validation shows strong QDE-to-human alignment (ρ=0.715), improving over ungrounded LLM-as-judge (ρ=0.668) and reference-based structural scoring (ρ=0.462). Experiments on eight frontier configurations show that agentic refinement with Claude Opus 4.6 approaches the human-authored gold baseline, while residual failures concentrate in topology preservation, label grounding, and type-specific syntax. Because QDE requires no reference at inference time, it is immediately deployable as an evaluator for diagram assistants and as a reward signal for fine-tuning diagram agents. We hope these resources support progress on reliable diagram generation and, more broadly, on evaluation methods for structured visual code in production LLM systems.

Limitations

MERMAIDBENCH covers Mermaid only; results may not transfer directly to PlantUML, DOT, or TikZ, though Mermaid is the most widely adopted LLM-native diagram language and its typed grammars produce structured, parseable output particularly suited to fine-grained automated evaluation. We cover Mermaid's 11 stable core types and exclude the trivial `pie` type, five beta-tagged types with unstable syntax (`sankey-beta`, `xychart-beta`, `block-beta`, `packet-beta`, `architecture-beta`), plugin-based extensions (`C4Context`, `zenuml`), `kanban` (added post-corpus collection), and `requirementDiagram` (extremely niche). The source corpus is English-heavy and web-scraped, so domain skew is possible; we release only distilled Mermaid golds and metadata where licensing prohibits redistribution of raw images. QDE Level 1 is fully deterministic, but Level 2 still uses a multimodal LLM at inference; grounding and k=5 self-consistency reduce variance and judge bias but do not eliminate them, and the input-side verbosity leniency described in §7 means absolute QDE scores are not perfectly comparable across tiers.

Disclaimer

This paper was prepared for informational purposes in part by the Machine Learning Center of Excellence group of JPMorgan Chase & Co. and its affiliates ("JP Morgan") and is not a product of the Research Department of JP Morgan. JP Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained

herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021. Program synthesis with large language models. *Preprint*, arXiv:2108.07732.

Jonas Belouadi, Anne Lauscher, and Steffen Eger. 2024. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2024. SWE-bench: Can language models resolve real-world GitHub issues? In *Proceedings of the Twelfth International Conference on Learning Representations*.

Klaus Krippendorff. 2011. Computing Krippendorff's alpha-reliability. *Departmental Papers (ASC)*.

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-Tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. DS-1000: A natural and reliable benchmark for data science code generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18319–18345. PMLR.

Kaixin Li, Yuchen Tian, Qisheng Hu, Ziyang Luo, Zhiyong Huang, and Jing Ma. 2024. MMCode: Benchmarking multimodal large language models for code generation with visually rich programming problems. *Preprint*, arXiv:2404.09486.

Chumeng Liang and Jiaxuan You. 2025. Evaluating LLM-generated diagrams as graphs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12678–12690, Suzhou, China. Association for Computational Linguistics.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2263–2279. Association for Computational Linguistics.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A comprehensive overview of large language models. *Preprint*, arXiv:2307.06435.

Basel Shbita, Farhan Ahmed, and Chad DeLuca. 2025. MermaidSeqBench: An evaluation benchmark for NL-to-Mermaid sequence diagram generation. *arXiv preprint arXiv:2511.14967*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508. Association for Computational Linguistics.

Jingxuan Wei, Cheng Tan, Qi Chen, Gaowei Wu, Siyuan Li, Zhangyang Gao, Linzhuang Sun, Bihui Yu, and Ruifeng Guo. 2024. From words to structured visuals: A benchmark and framework for text-to-diagram generation and editing. *arXiv preprint arXiv:2411.11916*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Abhay Zala, Han Lin, Jaemin Cho, and Mohit Bansal. 2024. DiagrammerGPT: Generating open-domain, open-platform diagrams via LLM planning. In *Proceedings of the Conference on Language Modeling (COLM)*.

Yi Zhao, Zhen Yang, Shuaiqi Duan, Wenmeng Yu, Zhe Su, Jibing Gong, and Jie Tang. 2025. PlotGen-Bench: Evaluating VLMs on generating visualization code from diverse plots across multiple libraries. *Preprint*, arXiv:2601.11525.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin,

Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623. Curran Associates, Inc.

A Additional MermaidBench Details

The main paper summarises MERMAIDBENCH at a high level. This appendix gives the supporting construction detail needed to reproduce the benchmark and interpret the aggregate statistics. Figure 4 illustrates the design principle that difficulty is introduced through the user query while the gold Mermaid program remains fixed for a source diagram, isolating query-understanding difficulty from variation in target-code complexity.

Tier generation. Every verified baseline participates in every tier. T1 directly names diagram elements; T2 introduces adversarial noise, hedging, and distractors around the same requirements; T3 uses workflow/business phrasing without Mermaid terminology; T4 describes an indirect scenario requiring structural inference. Multi-turn tasks begin from a stripped-down baseline (edges and styling removed) and ask for specific edits toward the full gold. The gold Mermaid code is held fixed across single-turn tiers so that scoring differences reflect query-comprehension difficulty rather than gold-side variation.

Quality gates. Baselines pass three gates before release: (i) syntax/render validation using a pinned Mermaid.js runtime; (ii) canary-token preservation (three `XQV{NNNN}_{shortid}` mutations per baseline, verified after mutation); and (iii) readability thresholds—label overlap < 0.08, whitespace ratio $\in [0.30, 0.85]$, aspect ratio $\in [0.3, 3.0]$—plus type-specific extractability. Baselines failing any gate are repaired or excluded.

Source collection and VLM-assisted authoring. Sources were collected from academic papers, technical documentation, and image search, then filtered by an LLM-first classifier and simple structural gates. From 323 verified source images, 230 quality-filtered, type-balanced sources entered distillation. For each source image, the authoring pipeline first asks a vision model (Claude Opus, selected after calibration against GPT-4o/GPT-5.2) to describe the components, relationships, and layout, then asks it to generate Mermaid conditioned on both the image and the description. A rendering-based retry loop supplies fidelity-guided feedback when type-specific structural checks fail (e.g. node/edge count mismatches for flowcharts, participant/message mismatches for sequence diagrams).

Human audit and splitting. The VLM draft is not treated as ground truth. All 230 baselines undergo expert audit for type fidelity, element coverage, structural accuracy, hallucinations, and label faithfulness; identified errors are repaired before release. The 70/30 dev/test split is stratified by type and tier and includes an anti-leakage constraint that prevents variants of the same source from crossing splits.

Type	Base	T1–T4	MT	Dev	Test
flowchart	24	96	24	85	35
sequenceDiagram	28	112	28	100	40
classDiagram	24	96	24	85	35
stateDiagram	18	72	18	65	25
erDiagram	19	76	19	65	30
gantt	24	96	24	85	35
gitGraph	27	108	27	95	40
journey	9	36	9	30	15
mindmap	20	80	20	70	30
quadrantChart	15	60	15	55	20
timeline	22	88	22	75	35
Total	**230**	**920**	**230**	**810**	**340**

Table 7: Full dataset composition. MT = multi-turn tasks. Dev/test split is stratified by tier and type with source anti-leakage.

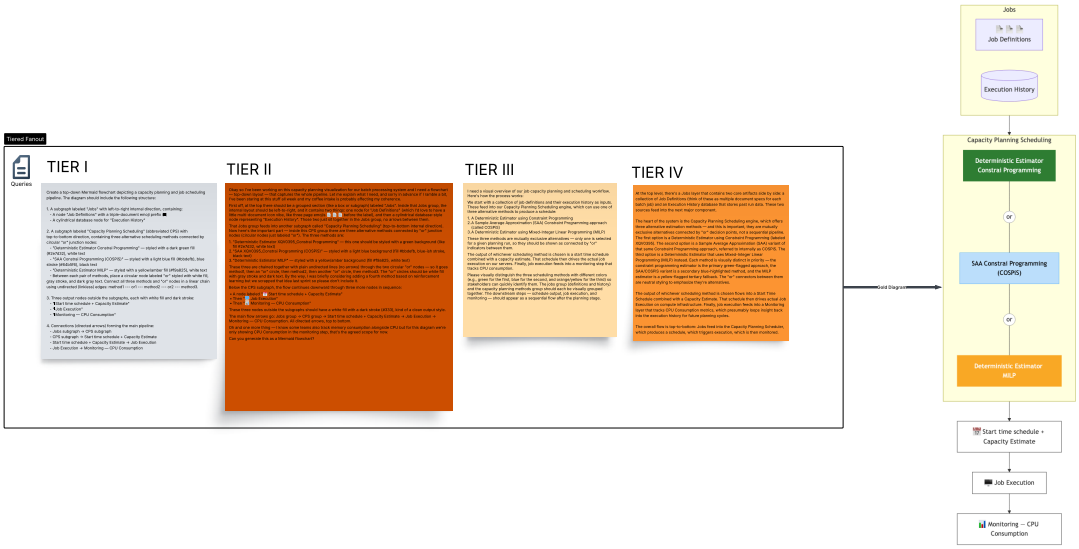


Figure 4: Example tier progression for a single source diagram. All four query tiers share the same gold Mermaid code; difficulty is controlled entirely on the query side. T1 names elements explicitly; T4 introduces noise and indirection.

A.1 Canary Contamination Audit

Canary identifiers guard against a model having memorized a test gold. Because a naive "canary appeared in the output" test is confounded by input copying, we specify the mechanism and detection precisely and report an audit over every evaluated system.

What is mutated, and where canaries can appear. Canaries are injected into *gold code node IDs*: during construction we rename three recurring identifiers per baseline to globally unique tokens `XQV{NNNN}_{shortid}` (e.g. `User`→`XQV0323_User`), rewriting every occurrence in the gold. The tier queries, however, are authored *from* the mutated gold, so canary strings routinely leak into the query text: in the 340-task test split, 183/340 (54%) golds carry at least one canary and 137/340 (40%) queries expose a canary token (full `XQV0323_User` or bare `XQV0323` prefix); restricted to canary-bearing golds, 63% expose the canary in the query. For multi-turn tasks, the simplified starting diagram supplied to the model likewise contains canary IDs. Consequently canaries are *not* hidden from the model, and their appearance in a generation is expected copying, not evidence of memorization.

Which canaries are hidden from the model. The only inputs guaranteed canary-free are the public *dev* tasks, from which canaries are reverted to their original identifiers before release; the *test* split retains canaries so that a memorized test gold would surface a token the model was never shown. Thus a canary functions as a contamination probe *only* on the subset of test tasks where it does not already appear in the query or conversation context.

Detection procedure. For each generation we extract exact matches of the token pattern `XQV`⟨4 digits⟩`_`⟨id⟩ and, for every match, check whether the same canary prefix appears in any model-visible input for that task (the query text plus, for multi-turn tasks, the full conversation history). A match whose prefix is present in an input is a *copyable* emission (echoing a provided ID) and is discarded; a match whose prefix appears in *none* of the visible inputs is a *hidden-canary* match and is the only signal we count as possible contamination.

Audit result. We ran this test over all nine evaluated configurations on the test split (Table 8). Copyable emissions are common—up to 135 tasks per model—and concentrate on tasks whose query or multi-turn context already contains the canary; the two newest Claude configurations rewrite labels and emit no canaries at all. Critically, **no configuration produced a single hidden-canary match**: every emitted canary was traceable to a model-visible input. Under this criterion we detect no memorization of test golds. We note two limitations:

the probe covers only the three mutated IDs per gold (not all tokens), and it detects verbatim reproduction rather than paraphrased recall; a stronger guarantee would additionally strip the residual bare-prefix canary leakage from test queries, which we leave to a future benchmark revision.

System	Tasks w/ can.	Copyable tok.	Hidden tok.
Claude Opus 4.6	129	330	**0**
Claude Opus 4.6 (reasoning)	130	333	**0**
Claude Opus 4.7	0	0	**0**
Claude Opus 4.7 (reasoning)	0	0	**0**
GPT-5.4 (no reasoning)	135	339	**0**
GPT-5.4 (reasoning)	120	307	**0**
MERMAIDAGENT/GPT-5.4	60	156	**0**
DIAGRAMAGENT/GPT-5.4	130	321	**0**
DIAGRAMAGENT/Opus 4.7	0	0	**0**

Table 8: Canary audit over the 340-task test split. *Tasks w/ can.* = number of tasks whose output contains at least one canary token; *Copyable tok.* = count of emitted canary tokens whose prefix appears in a model-visible input (query or conversation context); *Hidden tok.* = emitted canary tokens absent from all visible inputs (the contamination signal). No system produced a hidden-canary match.

B Additional QueryDiagramEval Details

Deterministic fact extraction. Table 9 lists the structural facts parsed for each Mermaid family. Extractors are type-specific because Mermaid types encode different semantics: a sequence diagram is primarily participants and messages, a Gantt chart is tasks and sections, and a git graph is branches and commits. Feeding these facts to the auditor reduces counting errors and discourages visual hallucination by making the code's actual contents explicit.

Diagram type	Extracted facts
flowchart	Nodes, edges, subgraphs
sequenceDiagram	Participants, messages with labels
classDiagram	Classes, relationships with types
erDiagram	Entities, relationships
stateDiagram	States, transitions
gantt	Tasks, sections, date format
gitGraph	Branches, commits
journey	Steps with satisfaction, sections
mindmap	Hierarchy tree
quadrantChart	Data points, axes, quadrant names
timeline	Events and time periods

Table 9: Type-specific structural facts extracted deterministically from candidate code (no LLM inference).

Parser accuracy audit. Because the fact block anchors the semantic auditor, its faithfulness to the candidate code matters directly. We audited the type-aware parser against an independent regular-expression oracle on all 340 human-verified gold diagrams of the resplit benchmark (68 sources $\times$ 5 difficulty tiers). For each diagram we compared the parser's element counts (participants, classes, entities) and its relation set (edges, messages, relationships) against the oracle; Table 10 reports per-type element-count exactness and relation recall for the five structural families.

The audit surfaced—and we fixed—several parser gaps that had depressed relation recall on two families: flowchart middle-label edges (`A -- label --> B`) and quoted/Unicode pipe labels (`S -->|"ε"| A`), which previously yielded zero edges; sequenceDiagram inline-activation shorthand (`A-»+B: msg`); classDiagram relationships carrying multiplicity labels (`A "1" -- "*" B`); and erDiagram entities declared through attribute blocks or bearing `:::style` suffixes. Before these fixes, relation recall was 71% (flowchart), 88% (sequenceDiagram), and 72% (classDiagram), and entity-count exactness was 83% (erDiagram). After the fixes, element counts and relation recall both reach 100% against the oracle across all five structural families (Table 10).

Two caveats apply. First, this is an automated cross-check against a second deterministic oracle rather than a human element-by-element audit; correlated errors shared by both parsers would not be detected, so a full human-verified precision/recall audit across all eleven types remains future work. Second, the fact block is *anchoring evidence*, not the sole basis for scoring: the auditor prompt instructs the judge to inspect the code and rendered image directly when a fact is absent, so an occasional missed relation degrades to ordinary multimodal reading rather than a silent scoring error, and the Level 1 gates independently catch total parse failures.

Type	n	Element-count exact	Relation recall
flowchart	35	—	100%
sequenceDiagram	40	100%	100%
classDiagram	35	100%	100%
erDiagram	30	100%	100%
stateDiagram	25	—	100%

Table 10: Parser accuracy audit against an independent oracle on 340 human-verified gold diagrams. *Element-count exact* is the fraction of diagrams whose extracted element count (participants/classes/entities) exactly matches the oracle; *relation recall* is the fraction of oracle relations (edges/messages/relationships) recovered. Dashes mark families for which the oracle does not provide an independent element-count reference.

Rubric weighting. Table 11 gives the five dimensions used by the fact-grounded auditor. Adherence and completeness receive the highest weights because diagram-generation failures are usually semantic; readability and accessibility remain in the score but with lower weight so that a visually polished diagram cannot compensate for failing to satisfy content requirements. Weights were fixed a priori by rubric design and were not tuned on the annotation data.

Dimension	w_d	Evaluates
Adherence to prompt	0.30	Scope, constraints, style
Completeness	0.25	Required components present
Conceptual correctness	0.20	Logical relationships
Readability	0.15	Spacing, crossings, sizing
Accessibility	0.10	Contrast, color independence

Table 11: Fact-grounded audit rubric dimensions and weights.

Compilation gates in detail. Before semantic scoring, QDE extracts Mermaid code from fenced blocks, parses it with a type-aware AST parser, and renders it with a pinned Mermaid.js runtime. Parse failure caps the score at 25; render failure at 40 (empty code at 20); diagram-type mismatch at 60 (or 85 if content is otherwise correct). These gates are deliberately separated from the LLM auditor so that non-renderable or wrongly typed code cannot receive a high score merely because it appears plausible in text.

Reported metric suite. Table 12 separates deterministic Level 1 diagnostics from the Level 2 QDE score. We report render success and type accuracy alongside QDE because they expose different failure modes: a model can score well semantically only after it produces valid, renderable, correctly typed code. All metrics in the table are reference-free.

Metric	Level	Det.	Ref.
Render success rate	1 (Compilation)	✓	—
Type accuracy	1 (Compilation)	✓	—
S_{fg} / QDE	2 (Semantic)	grounded	—

Table 12: QDE metric suite. "Grounded" denotes LLM auditing anchored by deterministic structural facts.

C Human Annotation Protocol

Because our central claim is that QDE aligns with human judgment, we document the annotation study in full: who annotated, how they were trained, what they saw, the exact rubric, adjudication, per-dimension agreement, and compensation. Annotation followed a fixed written guideline (v1.0, April 2026) issued to every annotator before work began.

Annotator pool and expertise. Annotations were produced by a contracted professional annotation team of six annotators (anonymised A01–A06). They were *technical but not diagram-generation experts*: the guideline required only that annotators be able to *read* Mermaid and judge whether a diagram satisfies a natural-language request—not to author Mermaid code. Baseline familiarity was established through a 30-minute Mermaid tutorial with ten worked examples and per-type syntax cheat-sheets covering all eleven diagram families. Three annotators (A01–A03) carried the full-coverage load (∼2,480 items each); the remaining three (A04–A06) rated stratified subsets (∼930 each).

Training and calibration. Every annotator completed a mandatory calibration phase before any production annotation: (i) all annotators independently rated the *same* 20 calibration tasks; (ii) a one-hour calibration meeting discussed and resolved disagreements against reference anchors; and (iii) annotators re-rated 5 tasks to confirm alignment. Annotators could not begin the main phase until calibration was complete. During the main phase, tasks were presented in randomised order stratified by tier and type, with enforced rest breaks to limit fatigue effects, and annotators were instructed not to discuss individual items with one another.

Blinding. The metric-validation annotations (Task C) are *reference-blind*: annotators rated each model prediction purely against the user query and never saw the gold diagram, the source model identity, or the automatic quality bucket the item was drawn from. Non-rendering outputs were excluded upstream so that annotators always judged a visible diagram. This isolates query-satisfaction judgments from gold leakage and model-identity bias.

Rubric. Each prediction was scored on seven 1–5 dimensions plus structured error fields (Table 13). Dimensions C1–C6 were designed to mirror the automated auditor's dimensions to enable per-dimension correlation; C0 (overall quality) was rated holistically as an independent gut judgment rather than a weighted average of C1–C6. Each dimension carried explicit $1/3/5$ anchor descriptions in the guideline. In addition, annotators selected all applicable tags from a 15-way error taxonomy (C7; the basis for Appendix F.6) and listed missing (C8) and spurious (C9) elements as free text.

ID	Dimension	Scale / content
C0	Overall quality	1–5 holistic
C1	Readability	1–5
C2	Accessibility	1–5
C3	Adherence & constraints	1–5
C4	Completeness	1–5
C5	Conceptual correctness	1–5
C6	Factual/structural accuracy	1–5
C7	Error tags	multi-select (15-way taxonomy)
C8	Missing elements	free-text list
C9	Spurious elements	free-text list

Table 13: Human annotation rubric (Task C, reference-blind). C1–C6 mirror the automated auditor dimensions; C0 is an independent holistic rating.

Adjudication and item repair. For ordinal ratings the median of the primary ratings was used; items whose ratings spanned more than two points were flagged for discussion. For binary judgments, unresolved disagreement among the primary annotators was escalated to a senior annotator who made the final call. Separately, benchmark query–gold pairs (Task B) flagged by ≥ 2 annotators as low-faithfulness (B1 ≤ 2) or insufficient (B2 $=$ No) were repaired—query rewritten or gold corrected—or discarded by the project lead before the splits were frozen and any model evaluation began.

Per-dimension agreement. Table 14 reports inter-annotator agreement for every rubric dimension, not only holistic quality. Agreement was computed over the multiply-annotated pool (297 items with four independent ratings; 1,800 rating pairs on the 1–5 scale). Semantic dimensions most tied to diagram structure show the strongest agreement—completeness (α=0.838), holistic quality (0.797), adherence (0.795), conceptual correctness (0.757)—while the more subjective visual dimensions (accessibility 0.514; structural fidelity 0.528) are lower, partly due to ceiling effects (several annotators assigned near-uniform high scores). Mean absolute deviation between annotators is 0.30–0.45 on the five-point scale, i.e. annotators typically agree within about half a point. Holistic-quality agreement is stable across the five batches ($\alpha \in [0.69, 0.84]$) and pairwise annotator Spearman ranges 0.59–0.88 (mean 0.759, 30 pairs).

ID	Dimension	α	ICC	MAD
C0	Overall quality	**0.797**	0.795	0.45
C1	Readability	0.613	0.612	0.30
C2	Accessibility	0.514	0.513	0.33
C3	Adherence	0.795	0.793	0.45
C4	Completeness	**0.838**	0.837	0.33
C5	Conceptual correctness	0.757	0.755	0.31
C6	Structural fidelity	0.528	0.528	0.34

Table 14: Inter-annotator agreement per rubric dimension on the Task C pool (Krippendorff's α, intraclass correlation, and mean absolute deviation on the 1–5 scale). Computed over 297 items with four independent ratings.

Compensation and ethics. Annotation was carried out by a contracted professional annotation team; annotators were compensated by their employer under the terms of that commercial engagement rather than per-task by the authors. Annotators rated only synthetic or model-generated diagram content paired with authored queries and were not asked to provide personal or sensitive information.

D Fact-Grounded Judge Prompts

This appendix reproduces the prompts used by the fact-grounded semantic-assessment stage of QUERYDIAGRAMEVAL (S_{fg}, §3). The pipeline issues one system prompt (§D.1) plus a per-task user message that bundles the query, candidate code, deterministic structural facts (§D.2), and the rendered PNG (§D.3). k=5 independent samples are drawn from the judge with temperature 1.0 and aggregated by the rule of §3.2.

D.1 System prompt

System prompt (fact-grounded judge)

You are an expert diagram quality evaluator. You will assess how well a generated Mermaid diagram satisfies a user's request.

You are given:
1. The user's original query
2. The generated Mermaid code
3. A rendered image of the diagram (if available)
4. **Deterministic structural facts** extracted from the code — these are ground truth about what the code actually contains

Use the structural facts as anchoring evidence while performing semantic quality assessment.

Evaluation Process

Step 1: Requirement Extraction. Read the user query and enumerate every atomic requirement (R1, R2, . . .). Aim for 10–30 granular requirements.

Step 2: Fact-Anchored Requirement Checking. For each requirement, check whether it is satisfied using BOTH:
- The structural facts (element lists, counts, type detection)
- Your own semantic analysis of the code and image

For each Rn, output one of {satisfied, missing, partial}. When the structural facts confirm an element exists, treat that as strong evidence. When they don't mention something, check the code and image yourself.

Step 3: Hallucination Detection. Check for elements in the diagram NOT requested. Use the structural facts to systematically verify each element has a corresponding requirement.

Step 4: Visual Quality Assessment. Score readability and accessibility from the rendered image (or code heuristics if no image).

Step 5: Structured Output. Return a JSON object with these exact fields:

```
{
  "requirements": [{"id": "R1", "text": ". . . ", "status": "satisfied|missing|partial", "evidence": "structural_fact|semantic|image"}],
  "hallucinated_elements": [". . . "],
  "error_counts": {"n_total_requirements", "n_satisfied", "n_missing", "n_partial", "n_hallucinated"},
  "readability": <0-5>, "accessibility": <0-5>,
  "adherence_to_prompt_and_constraints": <0-5>,
  "completeness": <0-5>, "conceptual_correctness": <0-5>,
  "explanations": [". . . "]
}
```

Scoring Rubric (apply mechanically from error counts; abbreviated)

adherence_to_prompt_and_constraints: 5 = 0 missing/partial, 0 hallucinated; 4 = 1 missing or 1–2 partial; 3 = 2 missing or 3+ partial; 2 = 3+ missing; 1 = major scope violation; 0 = completely wrong.
completeness: 5 = all satisfied; 4 = 1 missing; 3 = 2 missing; 2 = 3–4 missing; 1 = 5+ missing or <50% satisfied; 0 = <25% satisfied.
conceptual_correctness: 5 = all logic correct; 4 = 1 wrong direction or minor logic error; 3 = 2 logic errors; 2 = fundamental misunderstanding; 1 = mostly wrong; 0 = completely wrong.
readability: 5 = clean layout; 4 = minor spacing; 3 = some overlap; 2 = significant issues; 1 = hard to read; 0 = unreadable.
accessibility: 5 = good contrast and labels; 4 = minor contrast issue; 3 = some concerns; 2 = poor contrast or tiny text; 1 = major issues; 0 = inaccessible.

D.2 Structural-fact summary

The structural-fact block is generated deterministically from the candidate code by a typed Mermaid parser; the auditor never sees the gold code. The block always includes parse status, render status, the detected diagram type, and code length; the remainder is type-specific.

Structural-fact template (verbatim)

```
STRUCTURAL FACTS (deterministic,
ground-truth from code analysis):
  Parse:   OK | FAILED
  Render:  OK | FAILED
  Detected diagram type: <type>
  Code length: <n> lines

  # Type-specific block (excerpt):
 Nodes (k): id1=label1, id2=label2, ...
  Edges (k): src -> dst [label]
  Sections / classes / participants /
    states / entities / events: ...
```

D.3 Per-task user prompt

Per-task user message

```
## User Query
<query>

## Generated Mermaid Code
```mermaid
<predicted code>
```

STRUCTURAL FACTS (deterministic, ...)
<fact summary from App. C.2>

Evaluate this diagram against the user's
requirements. Use the structural facts as
```

```
ground truth for what the code contains.
Return ONLY the JSON object.

[image: <...>]
```

D.4 Aggregation

For every S_{fg} call, the auditor's per-dimension 0–5 ratings are mapped to a 0–100 score via the fixed rubric weights $w = (0.15, 0.10, 0.30, 0.25, 0.20)$ over (readability, accessibility, adherence, completeness, conceptual correctness). The final per-task QDE is the arithmetic mean of $k = 5$ independent samples; per-task standard deviation $\bar{\sigma}_k$ is reported in Table 2. Level 1 hard gates (parse/render/type) cap the score before averaging (§3.1).

E MermaidAgent Architecture

This appendix expands the main-paper description of MERMAIDAGENT (§5) into the full architecture used for the agentic rows of Table 2. The DiagramAgent baseline (Wei et al., 2024) is ported to the same pinned Mermaid backend with its original tool interface preserved, providing an apples-to-apples agentic comparison.

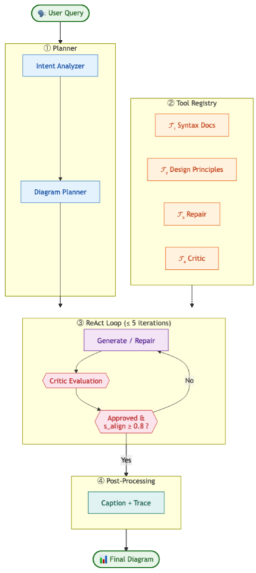


Figure 5: MERMAIDAGENT overview. A user query flows through the Planner (intent analysis and diagram plan), is grounded by four tools ($\mathcal{T}_1$–$\mathcal{T}_4$), and enters an iterative ReAct loop that generates, critiques, and repairs candidate diagrams until the quality threshold ($s_{\text{align}} \geq 0.5$) is met.

Planner. Given a user query, the Planner executes two sequential LLM calls under structured output schemas. The *intent analysis* call maps the query to a closed-vocabulary diagram type (one of the eleven supported Mermaid varieties), estimated complexity, subject domain, ranked key requirements, detected ambiguities, and style constraints. The closed vocabulary bounds the search space and prevents hallucination of unsupported types. The *diagram plan* call then consumes the intent analysis and produces (i) a structural outline, (ii) key nodes/entities, (iii) edges/relationships, (iv) an optional styling approach, and (v) a justification of how the plan meets the complexity requirements. The plan is injected into the generation prompt as a *soft* constraint, not mechanically enforced, so that generative flexibility is preserved while intent alignment is maintained.

Tool registry. Four tools support the agent: two *knowledge tools* consulted once per request and two *action tools* invoked within the ReAct loop.

Documentation ($\mathcal{T}_1$). Given a diagram type, returns a 700–1,000-character authoritative syntax summary covering keywords, edge syntax, node shapes, and structural constraints. For unknown types it returns the full list of supported types, enabling graceful degradation. This grounding prevents the LLM from hallucinating syntax from outdated pre-training data.

Design principles ($\mathcal{T}_2$). Maintains a static, expert-curated knowledge base of over 100 design heuristics along four axes: (i) *general* principles—clarity, consistency, simplicity, readability (4 guidelines each); (ii) *type-specific* rules for all eleven diagram varieties; (iii) *styling*—colour usage, shape semantics, annotation placement; and (iv) *complexity-level* element-count targets for simple (3–7), intermediate (8–15), and advanced ($\geq$ 16) diagrams. Per request, one query retrieves type-specific and general principles, and another retrieves complexity-level guidelines; both are injected into the generation prompt.

Repair ($\mathcal{T}_3$). Provides three utilities: *syntax validation* via a two-stage pipeline (lightweight structural checks for missing keywords, empty bodies, unbalanced delimiters, followed on pass by a full grammatical parse); *structure analysis* that counts diagram elements and identifies the diagram type; and *fix suggestion* via rule-based pattern matching on validation errors and structural counts to produce a prioritised list of repair suggestions.

Critic ($\mathcal{T}_4$). Performs a deterministic, multi-dimensional evaluation in five stages: (1) *syntax validity* (a syntax error classifies the diagram as poor quality and halves the alignment-score weight); (2) *type correctness* (a mismatch is recorded as a structural issue); (3) *requirement coverage* (Eq. 3); (4) *structural completeness* (diagrams with < 2 elements are flagged as too simple; those with > 50 elements trigger a decomposition recommendation; flowchart and sequence diagram types with zero edges are additionally flagged); and (5) *alignment scoring* (Eq. 4). For each key requirement r_i, the evaluator checks whether any content-bearing token (length > 3) appears in the lowercased diagram source:

$$s_{\text{cov}} = \frac{\left|\{r_i \in R \mid r_i \text{ is covered}\}\right|}{|R|}. \tag{3}$$

A coverage score below 0.5 triggers an improvement suggestion. The alignment score combines coverage with a syntax-weighted multiplier:

$$s_{\text{align}} = \text{clip}\big(s_{\text{cov}} \cdot w_{\text{syn}},\ 0,\ 1\big), \tag{4}$$
$$w_{\text{syn}} = \begin{cases} 0.7 & \text{syntax valid,} \\ 0.3 & \text{otherwise.} \end{cases}$$

The approval decision is the conjunction

$$\text{approved} \iff \text{syntax\_valid} \ \wedge\ \text{type\_match} \ \wedge\ |\text{design\_issues}| \le 1 \ \wedge\ s_{\text{cov}} \ge 0.7. \tag{5}$$

ReAct loop. The core loop runs up to $N_{\max} = 5$ iterations, each following a Thought $\rightarrow$ Act $\rightarrow$ Observe cycle. *Iteration 1* (initial generation) composes a prompt from the user query, intent analysis, diagram plan, syntax documentation ($\mathcal{T}_1$), and design principles ($\mathcal{T}_2$); the LLM returns Mermaid source and a brief explanation via a structured output schema, with markdown fences stripped automatically. Structural validation is applied immediately, and on failure a focused *syntax-only repair* sub-routine fires before critique. *Iterations* 2–$N_{\max}$ (repair) feed the LLM the current diagram, full critic feedback, user query, intent analysis, and original plan, and produce a repair plan listing issues, strategy, and concrete changes; this is followed by an optional syntax-only pre-repair and then a full repair whose prompt includes the repair plan and the first 1,000 characters of the syntax reference as a grounding guard. In syntax-only mode the LLM is constrained to minimal edits—no structural redesign, preservation of node IDs, labels, and edges unless they directly cause the error—and outputs raw Mermaid markup with a recorded change list.

Termination and post-processing. After each iteration, $\mathcal{T}_4$ evaluates the diagram. The loop exits early when approved $=$ true and $s_{\text{align}} \ge \tau$, with $\tau = 0.5$ by default; otherwise the last diagram is returned with a warning. Every step appends a record (action, output length, validity, alignment score, approval, issues addressed) to a trace log. Once the loop terminates, the agent prompts the LLM to produce a 2–3 sentence caption. The caption, final Mermaid source, iteration count, alignment score, and full trace are returned as the agent's structured response.

F Additional Experimental Results

The main paper reports the headline comparison across model configurations. This section restores supporting analyses: dataset separation, per-type behaviour, fact-grounding ablations, tier effects, judge-model choice, cross-judge sensitivity, human error tags, and a qualitative example.

F.1 Per-Type Analysis

Table 15 breaks QDE down by diagram family. The pattern is heterogeneous: Claude-based systems are consistently strong across many types, while GPT reasoning produces isolated wins on state diagrams and quadrant charts. MERMAIDAGENT/Claude Opus 4.6 is the most broadly competitive, leading on flowcharts, ER diagrams, and journeys and staying close to the best model on Gantt charts. The hardest families are gitGraph and classDiagram for several configurations, consistent with their dense symbolic syntax and sensitivity to relation direction, branching, and class-edge semantics. The gold row is a strong human reference rather than a hard ceiling because QDE scores query satisfaction rather than exact agreement with the gold program.

Model	flow	seq	class	state	ER	gantt	git	jour	mind	quad	time
GPT-5.4	65.7	79.5	83.0	70.1	85.8	86.6	56.2	83.3	69.1	62.8	84.4
GPT-5.4 (reasoning)	57.9	88.2	63.5	**92.0**	74.1	88.4	68.3	87.6	65.0	**97.1**	78.2
Claude Opus 4.6	86.0	87.0	88.0	83.6	90.3	**94.2**	70.4	89.6	**89.2**	82.3	83.1
Claude Opus 4.6 (reason.)	86.1	**93.6**	84.9	67.5	86.7	89.5	**86.0**	89.4	82.3	83.8	86.3
DiagramAgent w/ GPT-5.4	78.4	88.2	76.2	79.6	81.7	83.7	68.1	88.0	70.3	60.1	76.9
DiagramAgent w/ Claude	81.2	84.3	79.4	77.1	83.4	85.0	70.5	86.8	74.2	65.0	79.3
MermaidAgent w/ GPT-5.4	65.5	75.7	61.5	69.3	75.0	80.5	76.3	83.9	65.4	77.4	**86.7**
MermaidAgent w/ Claude	**89.5**	87.2	86.6	85.4	**90.5**	94.0	79.8	**90.0**	86.2	89.3	79.2
Gold baseline	88.4	89.7	**88.5**	84.5	87.6	93.8	79.4	87.8	80.8	90.4	82.7

Table 15: QDE by diagram type on the test set, averaged across tiers.

F.2 Metric and Judge Ablations

Table 16 decomposes the validation result by human rating dimension. Correlations are strongest with adherence, conceptual correctness, and completeness—the dimensions most tied to structural diagram semantics. Readability and accessibility are positive but weaker, reflecting that human holistic scores weight semantic fidelity more heavily than visual polish.

Human dim.	ρ(dim, S_{fg})	95% CI
C1 (Readability)	0.316	$[0.196, 0.426]$
C2 (Accessibility)	0.256	$[0.139, 0.371]$
C3 (Adherence)	0.699	$[0.620, 0.772]$
C4 (Completeness)	0.582	$[0.482, 0.668]$
C5 (Conceptual)	0.654	$[0.564, 0.733]$
C0 (Holistic)	**0.715**	$[0.632, 0.784]$

Table 16: Spearman ρ between each human rating dimension and the holistic S_{fg} score on the validation pool ($N = 255$; fact-grounded judge Claude Opus 4.6).

Quality bucket (C0)	n	ρ(C0, QDE)
Poor ($C0 < 2.5$)	24	0.244
Medium ($2.5 \leq C0 < 3.5$)	29	0.146
Good ($C0 \geq 3.5$)	202	0.567
All	255	**0.715**

Table 17: Per-bucket correlation stability on the validation pool (N=255). Buckets are defined on the holistic human scale (C0 ∈ [1, 5]): Poor < 2.5, Medium $\in [2.5, 3.5)$, Good ≥ 3.5. Within-bucket ρ is attenuated by range restriction and by a coarse, ceiling-heavy human scale ($69/255$ items rated exactly 5.0). QDE is most discriminative for broad quality separation—reliably distinguishing bad from good diagrams—which is the property most relevant to benchmark ranking.

Broad quality discrimination. Table 17 shows that QDE reliably separates poor from good diagrams, the property most relevant for benchmark ranking. Within-bucket correlations are lower, but this is expected: the human C0 scale is coarse and heavily tied at the ceiling (69 items at exactly 5.0), which mechanically depresses within-bucket Spearman independently of QDE's true resolution. The conclusions in Table 2 are also system-level means over 340 tasks rather than per-item rankings; aggregation averages out item-level noise, and the cross-judge study (Appendix F.5) confirms the resulting system ordering is stable (Kendall τ=0.905, top four systems rank-identical).

Configuration	ρ(C0)
S_{cot} (vanilla judge, no facts)	0.668
S_{fg} (generic facts)	0.688
S_{fg} (type-specific facts, $k = 5$)	**0.715**

Table 18: Fact-grounding ablation on the validation pool ($N = 255$). Type-specific structural facts contribute the largest single improvement ($+0.027$); generic structural facts add $+0.020$.

Fact-grounded judge	ρ(C0)	$\bar{n}_{\text{err}}$ (std)
GPT-4.1	0.568	10.1 (12.3)
GPT-5.2	0.595	6.3 (6.7)
Claude Opus 4.6	**0.715**	5.1 (6.0)

Table 19: Judge model comparison on the validation pool. Claude Opus 4.6 achieves the strongest calibration and is used as the default judge.

F.3 Reference-Based Structural Baseline (GRAPHF1)

The reference-based structural score reported in Table 3 (ρ=0.462) is GRAPHF1, a fully deterministic (no-LLM) metric that scores a candidate diagram against its gold Mermaid code. It is an *in-house reimplementation inspired by* the graph-based diagram metric of Liang and You (2025), not a wrapper around their released code; differences are itemized at the end of this subsection. We specify it in full so it is reproducible.

Graph abstraction. Both the gold and candidate code are parsed by the same typed Mermaid parser used for the structural facts (§D.2) and reduced to a directed graph $G = (V, E)$, where V is a set of labeled nodes and $E \subseteq V \times V$ is a set of ordered (src, dst) edges. $\text{GRAPHF1} = 0.4 \cdot \text{NODEF1} + 0.6 \cdot \text{EDGEF1}$, weighting relational structure above node inventory.

Label normalization. Each node label is lowercased, outer whitespace stripped, the characters _ - / \ mapped to spaces, and internal whitespace runs collapsed to single spaces. Two labels match *exactly* when their normalized forms are identical;

otherwise a *fuzzy* match fires when the normalized Levenshtein similarity $1-\text{editdist}/\max(|a|,|b|) \geq 0.8$. Edge style (solid vs. dashed), relationship keywords, cardinality annotations, and edge labels are discarded—only endpoints and direction are compared.

Node alignment. Nodes are matched one-to-one by a greedy pass in three priority tiers: (1) exact normalized-label equality; (2) fuzzy Levenshtein similarity ≥ 0.8; (3) a fallback on normalized node *IDs*, which recovers cases where a model reuses gold identifiers but omits labels. Each gold node aligns to at most one candidate node. Matched pairs are true positives; unmatched gold nodes are false negatives; unmatched candidate nodes are false positives, yielding NODEF1.

Edge direction. Edges are ordered (src, dst) pairs, so *direction is scored*: a reversed edge is simultaneously a false positive and a false negative. Candidate edges are re-expressed in the gold node space via the node alignment (the raw candidate endpoints are also retained, to catch direct ID reuse) and then set-intersected with the gold edge set to compute EDGEF1. Undirected relations (e.g. some class/ER associations) are stored in their parsed textual order and thus treated as directed for scoring purposes—a known conservative approximation.

Non-graph diagram types. The parser maps all eleven families to the (V, E) abstraction, but coverage is uneven and this is a deliberate limitation of the baseline. The seven graph-structured families populate both sets: flowchart (nodes/edges), `classDiagram` (classes / relationship edges), `erDiagram` (entities / relationship edges), `sequenceDiagram` (participants / message edges), `stateDiagram` (states / transition edges), `gitGraph` (commits / branch–commit edges), and `mindmap` (the one hierarchical list type retained as a full graph via parent–child edges). The remaining list-structured families collapse to a label set with an *empty* edge set, so on these types GRAPHF1 degenerates to NODEF1 over labels: `gantt` and `journey` reduce to their task/section labels (dates, durations, ordering, and journey scores are ignored), `timeline` to its period/event labels (chronological order ignored), and `quadrantChart` to its point labels (x/y coordinates and quadrant placement ignored). This structural blindness on non-graph types is one reason GRAPHF1 correlates far more weakly with human ratings than the fact-grounded auditor, which consumes type-specific facts (§D.2) that capture exactly these ordering, date, and coordinate signals.

Relation to Liang and You (2025). We adopt their central idea—node and edge F1 over a graph abstraction with fuzzy label matching—but the implementation is our own and differs in four ways: (i) we operate directly on parsed Mermaid source rather than on graphs extracted from rendered images; (ii) we use the normalization and three-tier alignment above rather than their embedding-based matcher; (iii) we fix the $0.4/0.6$ node/edge weighting; and (iv) we add type-specific parsing for Mermaid's non-flowchart families. Consequently GRAPHF1 is reported only as a reference-based structural yardstick for contextualizing QDE; its absolute values are *not* numerically comparable to the original DiagramEval scores.

F.4 Difficulty-Tier Effects

Table 20 reports changes between successive query tiers. The average pattern is not strictly monotonic: T1→T2 is nearly flat on average, while T2→T3 and T3→T4 are negative. Adversarial or noisy wording alone is less damaging than indirect descriptions requiring structural inference. The large drop for GPT-5.4 reasoning from T3 to T4 shows that extended reasoning does not automatically make Mermaid generation robust to distractors, whereas Claude reasoning is comparatively stable across tiers.

Model	ΔT1–2	ΔT2–3	ΔT3–4	ΔT1–4
GPT-5.4	+7.9	-2.2	-3.8	+1.9
GPT-5.4 (reasoning)	-6.2	+1.8	-11.2	-15.6
Claude Opus 4.6	+1.1	+3.3	-4.0	+0.4
Claude Opus 4.6 (reasoning)	-0.2	+0.2	-1.7	-1.7
DiagramAgent w/ GPT-5.4	+2.2	-7.1	-0.9	-5.8
DiagramAgent w/ Claude	+3.1	-5.5	-1.2	-3.6
MermaidAgent w/ GPT-5.4	-6.2	-1.8	+8.6	+0.6
MermaidAgent w/ Claude	-0.5	-2.9	-2.2	-5.6
Model mean	+0.2	-1.8	-2.1	-3.7

Table 20: QDE changes between successive query tiers; negative values indicate degradation.

F.5 Cross-Judge Sensitivity

To check whether the choice of fact-grounded judge meaningfully changes our conclusions, we draw

a stratified random sample of 175 (model, task) pairs (25 per system, 7 systems) and re-score every pair with GPT-4.1 as an alternative fact-grounded judge. Prompts, structural-fact construction, image attachment, and k=5 self-consistency are kept identical. Table 21 reports per-system means under each judge.

System	Claude	GPT-4.1	Δ	r_C/r_G
Claude Opus 4.6 (reasoning)	89.81	96.75	+6.94	1/1
MermaidAgent (Claude)	83.77	95.25	+11.49	2/2
Claude Opus 4.6	81.18	91.13	+9.94	3/3
GPT-5.4 (reasoning)	79.93	87.41	+7.48	4/4
GPT-5.4	75.70	81.86	+6.15	5/6
DiagramAgent (GPT-5.4)	74.36	83.78	+9.42	6/5
MermaidAgent (GPT-5.4)	68.58	77.25	+8.66	7/7

Table 21: Cross-judge sensitivity (n=25 per system, 175 total). Δ = mean QDE under GPT-4.1 minus mean under Claude; r_C/r_G = system rank under each judge.

GPT-4.1 is systematically more lenient (every system gains 6–11 points) but the relative ordering is highly stable: Kendall $\tau = 0.905$ (Spearman $\rho = 0.964$, Pearson $r = 0.966$, $n = 7$). The single rank swap involves two systems whose Claude means differ by only 1.34 points. Per-task agreement is $\rho = 0.698$ (Pearson $r = 0.927$, $n = 175$). Together with Table 19, this rules out a self-enhancement bias large enough to reorder the systems in Table 2.

F.6 Human Error Taxonomy

Table 22 gives the human error-tag distribution underlying the discussion in §7. The most common errors are missing edges, missing nodes, layout/readability problems, label mismatches, hallucinated nodes, and factual errors. Syntax and type mistakes are relatively uncommon, especially compared with semantic omissions, which is why render success and type accuracy are insufficient as standalone metrics.

Error type	T1	T2	T3	T4	MT	All
Syntax error	0.0	8.2	0.0	1.6	1.1	2.0
Type misclassification	4.6	4.9	1.3	0.0	1.1	2.2
Missing nodes	18.5	26.2	20.8	15.6	15.4	19.0
Missing edges	23.1	32.8	22.1	12.5	23.1	22.6
Hallucinated nodes	18.5	14.8	19.5	4.7	12.1	14.0
Hallucinated edges	15.4	8.2	13.0	10.9	11.0	11.7
Inverted logic	1.5	9.8	3.9	9.4	1.1	4.7
Label mismatch	16.9	18.0	18.2	15.6	7.7	14.8
Ordering error	9.2	11.5	9.1	6.2	6.6	8.4
Wrong cardinality	1.5	3.3	9.1	3.1	3.3	4.2
Over-specification	13.8	14.8	22.1	6.2	11.0	13.7
Layout/readability	18.5	23.0	11.7	14.1	11.0	15.1
Factual error	6.2	18.0	18.2	9.4	8.8	12.0

Table 22: Human error-tag distribution by tier (% of annotated predictions exhibiting each error; $n = 358$).

F.7 Qualitative Example

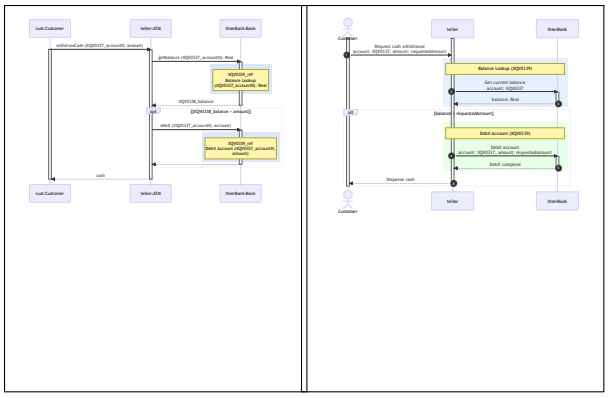
To make the failure modes in Tables 15 and 22 concrete, we walk through one held-out Tier-3 *sequenceDiagram* task in the rail-signaling domain (task `V4_T2_010526_180431_dd0d90`), representative of typed, workflow-oriented diagrams that appear in industrial documentation. The gold Mermaid program contains canary-mutated identifiers (`XQV03xx_*`) injected at construction time; the candidate is produced by MERMAIDAGENT backed by GPT-5.4 with no access to the gold.

User query (T3

I need a visual representation of our onboard equipment startup and connection procedure. Here's how the process works:

1. The train operator (user of the XQV0323 system) initiates an ignition request to the onboard Handheld Terminal (HHT).
2. Upon receiving the ignition request, the HHT performs an internal initialization sequence.
3. Immediately after initialization, the HHT runs a series of self-tests to verify its own operational readiness.
4. Once self-tests are complete, the HHT establishes a connection to the communication network (this is categorized under XQV0324 connectivity).
5. The HHT then sends a second request through the network specifically to connect to the Radio Block Centre (RBC).
6. The network relays the connection request onward to the RBC, formally requesting the establishment of that link on behalf of the operator.

The key participants are: the train operator, the HHT, the Network, and the RBC. Number the interactions so each step is referenceable. The HHT's initialization and self-test steps are internal and don't involve other parties.



(a) Gold rendering (held out) **(b) MERMAIDAGENT/GPT-5.4**

Figure 6: Rendered gold vs. prediction for the qualitative example. Both diagrams pass Level-1 gates; the candidate preserves participants and the six-step ordering but paraphrases canary-tagged labels and adds three `Note over` annotations.

Gold Mermaid (held out)

```
sequenceDiagram
    autonumber

    actor XQV0323_User
    participant HHT
    participant Network
    actor RBC

    XQV0323_User->>+HHT: XQV0322_Request ignition
    HHT->>HHT: Initialize
    HHT->>HHT: Start self-tests
    HHT->>Network: XQV0324_Connect to network
    HHT->>+Network: XQV0324_Connect to RBC
    Network->>RBC: XQV0322_Request the connection to RBC
```

MERMAIDAGENT w/ GPT-5.4 prediction

```
sequenceDiagram
    autonumber
    actor TO as Train Operator
    participant HHT as Handheld Terminal (HHT)
    participant NET as Communication Network
    participant RBC as Radio Block Centre (RBC)

    Note over TO,HHT: Startup initiated by the operator
    TO->>+HHT: 1. Ignition request

    Note right of HHT: Internal HHT processing only
    HHT->>HHT: 2. Internal initialization sequence
    HHT->>HHT: 3. Run self-tests for readiness

    Note over HHT,NET: XQV0324 connectivity
    HHT->>NET: 4. Establish connection to network

    Note over HHT,NET,RBC: XQV0324 RBC link establishment
    HHT->>NET: 5. Request connection to RBC via network
    NET->>RBC: 6. Relay RBC connection request

    deactivate HHT
```

What QDE captures. The candidate parses, renders, and is correctly typed, so all Level 1 gates pass. The fact extractor confirms that all four gold participants are present (`TO`/`HHT`/`NET`/`RBC`) and that the six-step ordering—ignition → self-tests → network → RBC—is preserved, which the fact-grounded auditor scores high on *conceptual correctness* and *completeness*. Two cost-bearing differences remain: (i) the candidate replaces canary-tagged labels (`XQV0322_*`) with descriptive paraphrases, and (ii) it adds three `Note over` annotations beyond the gold scope, which the auditor flags as soft hallucinations. The candidate also turns `RBC`'s `actor` role into a generic `participant`, a small type-specific drift that surface-only metrics would miss but that the structural facts surface explicitly. The example illustrates why QDE separates renderability from semantic fidelity: a high-quality but imperfect candidate receives strong adherence/completeness credit while still being penalised for label rewording and over-annotation, matching the dominant error categories in Table 22.

Robust Dual-Signal Fusion: Hybrid Neuro-Symbolic Gating with Compressed Chain-of-Thought Refinement for Irony Detection in Social Media Texts

Ankit Bhattacharjee, Krityapriya Bhaumik
Indian Institute of Technology Kharagpur, India
Correspondence: {ankit2005, ktpb124578}@kgpian.iitkgp.ac.in

Code:
https://github.com/SrabonGitikar/RDS-Fusion

Abstract

Small-scale Large Language Models (LLMs) natively default to literal semantic interpretations, making few-shot irony detection a persistent challenge in noisy, user-generated text. In this study, we show that, despite this limitation, inference from compact LLMs can still be effectively harnessed for this task. We introduce the **Robust Dual-Signal (RDS) Fusion** framework, a hybrid neuro-symbolic architecture that utilizes a compressed Chain-of-Thought (CoT) of an LLM, alongside a static, pretrained RoBERTa and a symbolic prior module in two downstream fusions. RDS Fusion demonstrates better performance than an unrestricted reasoning ablation, while requiring substantially lower inference time. We also find that in our experiments, unrestricted reasoning does not yield better results than restricted reasoning. Evaluated on a strictly held-out TweetEval test set ($N = 734$), RDS Fusion achieves 78.1% accuracy and a Macro F1 of 0.777, which is comparable to a finetuned BERTweet-base. On the heavily imbalanced iSarcasm dataset, the frozen CoT pipeline suppresses 22.5% of baseline false positives, yielding a Macro F1 of 0.6726 and Ironic F1 of 0.4821, outperforming multiple heavily supervised SemEval transformer ensembles. Paired McNemar's tests show that, while adding the symbolic prior to the neural baseline yields an insignificant gain, and the RDS Fusion is statistically insignificant compared to the combined RoBERTa and symbolic prior ablation; the concurrent fusion achieves a statistically significant improvement over the standalone baseline ($p = 0.005$).

1 Introduction

Detecting irony in unstructured social media text remains a persistent NLP challenge. Compact, open-weight LLMs natively default to literal interpretations, frequently missing the contextual inversions that define irony. In fact, standalone few-shot evaluation of `Qwen2.5-3B-Instruct` (Yang et al., 2024) on unstructured irony yields a poor performance (Macro F1 < 0.500), suggesting that small scale local models cannot function as primary detectors in this domain without structural grounding. While Chain-of-Thought (CoT) prompting enables models to map complex pragmatics (Wei et al., 2022), unconstrained CoT introduces severe computational latency for tasks involving chain-of-thought (Xia et al., 2025). Frameworks like TokenSkip (Xia et al., 2025) reduce inference overhead via compression of the chain-of-thought, but depend on supervised finetuning (SFT) for domain adaptation and are primarily validated on structured datasets (MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021)). However, the application of chain-of-thought and its compression are less explored in tasks like sentiment analysis and sarcasm detection.

While fine-tuning remains the standard paradigm for adapting LLMs to narrow classification tasks, it carries practical drawbacks. The objective of our framework is not to replace supervised fine-tuning, but to bypass the necessity for it. By demonstrating that inference-time neuro-symbolic reasoning can recover fine-tuned performance levels, we establish a mechanism to achieve task-specific parity while preserving the frozen base model's capabilities.

We identify two concurrent failure modes in naive CoT compression applied to irony: (1) a *literal bias* in few-shot LLMs that causes overestimation of literal intent, and (2) a *precision collapse* when explicit heuristic priors are added to maximize recall, as aggressive symbolic over-firing floods the system with false positives. Neither failure can be resolved in isolation.

We propose the **Robust Dual-Signal (RDS) Fusion** framework to resolve this tension. RDS Fusion is a hybrid neuro-symbolic architecture with

Proceedings of the 11th Workshop on Natural User-generated Text, pages 266–285
October 28, 2026 ©2026 Association for Computational Linguistics

three pillars: a pretrained RoBERTa (Liu et al., 2019) component, a linguistically-motivated symbolic prior for explicit irony detection, and a prediction by an LLM. We obtain the chain-of-thought given by the LLM at a fixed budget of tokens, and further, compress the CoT post-generation, by the particular mechanism discussed in 3.3.3. The exact role of the compression, the key difference from TokenSkip, and the motivation behind deriving a new *compressed CoT based score*, when the score assigned by the LLM is not present in the compressed CoT, are discussed in Appendix B.

We empirically demonstrate that, for RDS Fusion (with post-generation pruning), the symbolic prior is load-bearing for recall while the **compressed** CoT pipeline functions primarily as a *precision-recovery engine*, suppressing false positives generated by heuristic over-firing. However, an *unrestricted ablation* revealed that the original LLM inference does not recover precision in the same way as the compressed RDS Fusion pipeline. Rather, the LLM frequently predicts very high scores for tweets that are literal.

Primary Contributions. (1) A few-shot neuro-symbolic fusion framework that employs a rule-based selective post-generation token pruning, without SFT.
(2) An empirical reframing of few-shot reasoning in irony detection, showing the CoT pipeline acts primarily as a precision-restoration filter rather than a primary detector.
(3) Validation on TweetEval ($N = 734$), matching fine-tuned BERTweet-base (78.1% accuracy, Macro F1 0.777), with statistical significance ($p = 0.005$) to the baseline.
(4) Robust few-shot cross-domain generalization on iSarcasm without parameter modification, suppressing 22.5% of false positives and outperforming multiple supervised SemEval systems.

2 Related Work

2.1 Chain-of-Thought (CoT) Compression

CoT prompting decomposes complex reasoning into intermediate steps (Wei et al., 2022) but introduces quadratic attention overhead (Vaswani et al., 2017) and linear KV-cache growth. Compression approaches range from perplexity-based token filtering (Li et al., 2023) to bidirectional token importance classification, as in LLMLingua-2 (Pan et al., 2024). TokenSkip addresses this by applying quantile-thresholded compression over importance scores and utilizing Supervised Fine-Tuning (SFT) to teach LLMs to autonomously skip redundant tokens during autoregressive generation. However, the latency reduction in our framework is strictly due to the rigid threshold of `max_new_tokens`.

2.2 Neuro-Symbolic and Training-Free Inference

Inference-time compute scaling (e.g., OpenAI o1 (OpenAI, 2024b)) demonstrates that dynamic reasoning budgets improve performance without parameter updates. Hybrid neuro-symbolic architectures (d'Avila Garcez and Lamb, 2023; Kautz, 2022) combine neural continuous processing with symbolic discrete overrides, providing the structural grounding needed for explicit irony markers that neural pipelines smooth away (Riloff et al., 2013). RDS Fusion instantiates this paradigm for compressed CoT in social media irony.

3 Methodology

RDS Fusion employs a two-stage CoT compression process, consisting of a fixed generation-time token budget followed by a post-generation token pruning. Figure 1 illustrates the full pipeline. First, the LLM is constrained to generate at most 120 new tokens. This fixed budget limits the computational cost of autoregressive reasoning.

Second, the generated trajectory is subjected to task-aware post-generation pruning. The Local Guardian assigns k tokens as critical according to the gradient-based sensitivity of individual tokens to the RoBERTa classification objective. Tokens are classified into four sections, as discussed in subsection 3.3.3. The resulting compression therefore operates on the sequence actually generated within the fixed token budget, retaining the subset of tokens most relevant to the downstream irony decision, preserving their original order.

Together, these stages provide compression at both the generation and representation levels: the fixed generation budget bounds the length of the reasoning trajectory produced by the LLM, while post-generation pruning further reduces the information passed to the fusion layer. The resulting compressed trajectory is subsequently used to derive P_{CoT}, whose construction and relationship to the LLM's original confidence score are described in subsection 3.3.1.

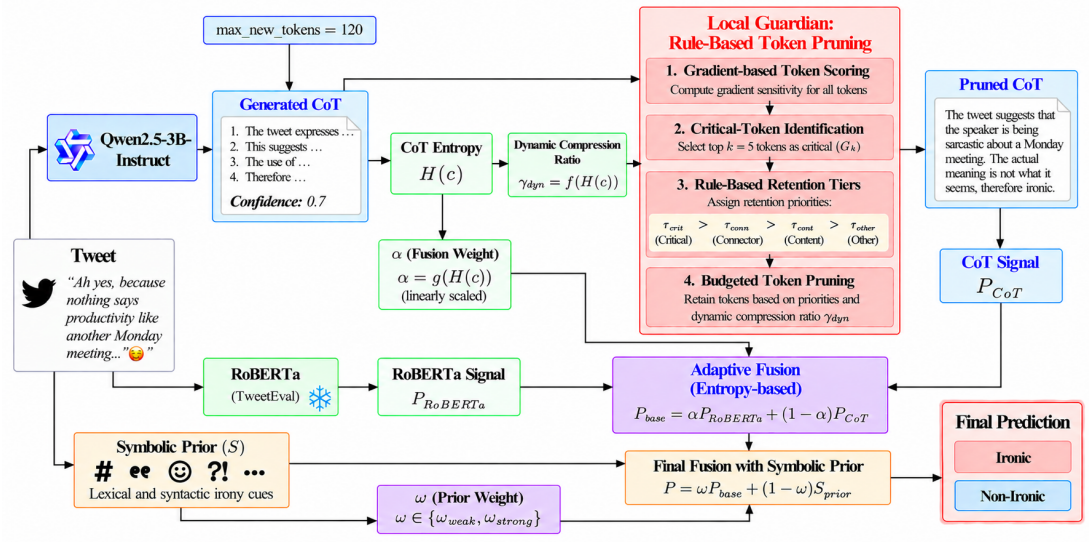


Figure 1: The RDS Fusion pipeline architecture.

3.1 Pretrained Neural Backbone

The neural component of RDS Fusion employs a pretrained RoBERTa-based irony classifier as its downstream backbone. This classifier is pre-trained on the training set of TweetEval Irony (Barbieri et al., 2020). The RoBERTa parameters remain **frozen**, with the model serving as a fixed neural signal (for any possible test set) that is subsequently combined with the symbolic prior and CoT-derived signal through the RDS Fusion mechanism. Further details regarding the pretrained model, inference configuration, and implementation are provided in subsection 4.3.

3.2 Few-Shot Prompt for Generation of LLM's Chain-of-Thought

The generative pipeline relies on a manually authored 4-shot Chain-of-Thought (CoT) prompt to evaluate irony, preventing data contamination from the evaluation sets. These examples represent four canonical boundary cases: pure sincerity, explicit sarcasm, contextual ambiguity, and symbol-driven irony (e.g., usage of hashtags, emojis etc).

The prompt structurally grounds the generator by defining irony as a pragmatic contrast and forcing the evaluation of key signals like exaggerated tone and real-world contradiction. Post-reasoning, the LLM outputs a continuous confidence score $P_{\text{CoT}}(\hat{y} = 1|c) \in [0, 1]$, mapping from absolute literal (0.0) to ironic (1.0) certainty. The full prompt template is provided in appendix B.1.

3.3 Post-Generation Token Pruning Mechanism

Given LLM $\mathcal{M}$ and CoT trajectory $c = \{c_i\}_{i=1}^{m}$, vanilla TokenSkip assigns importance $I(c_i)$ to each token and prunes below the γ-quantile threshold $I_\gamma = Q_\gamma(I(c_1), \ldots, I(c_m))$, retaining $\tilde{c} = \{c_i \mid I(c_i) \geq I_\gamma\}$.

3.3.1 Prediction from Compressed CoT

RDS Fusion derives its reasoning signal from the compressed trajectory rather than assuming that the LLM's original numerical confidence is preserved. Let c denote the trajectory generated under the fixed `max_new_tokens=120` budget and $\tilde{c}$ its post-pruned form. We define $P_{\text{CoT}} \in [0, 1]$ as the continuous irony estimate extracted from $\tilde{c}$. If an explicit score survives compression it is extracted directly; otherwise, the original score is discarded and P_{CoT} is inferred from the surviving reasoning. Thus, P_{CoT} is a compression-derived estimator and need not equal the confidence originally expressed by the LLM. The extraction uses explicit score/verdict matching followed by negation-aware reasoning-based heuristics, with 0.5 denoting unresolved ambiguity and triggering a 10-token forced-verdict fallback. This process in discussed in detail in Appendix B.4.

3.3.2 Predictive Entropy for Budgeting and Inference Confidence

We compute the token-level predictive entropy, $H(c)$, to serve a dual architectural mandate: quantifying the LLM's internal confidence to regulate downstream adaptive fusion (Section 3.5), and act-

ing as a substitute for the dynamic token allocation of Supervised Fine-Tuning (SFT).

To avoid the rigidity of compression, we introduce a dynamic ratio γ_{dyn} calibrated via $H(c)$:

$$H(c) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{v\in\mathcal{V}} P(v|\mathbf{c}_{<i})\log P(v|\mathbf{c}_{<i}), \tag{1}$$

where m is the generation length, $\mathcal{V}$ is the model's vocabulary, and $\mathbf{c}_{<i}$ is the context sequence prior to step i. High entropy indicates structural ambiguity and therefore motivates a more aggressive compression budget. We map $H(c)$ to γ_{dyn} via a scaled sigmoid:

$$\gamma_{\text{dyn}} = \gamma_{\min} + \frac{\gamma_{\max} - \gamma_{\min}}{1 + \exp[-k(H(c) - \mu)]}, \tag{2}$$

where μ is the empirical mean entropy and k controls sensitivity. Bounds $(\gamma_{\min}, \gamma_{\max}) = (0.1, 0.9)$ prevent degenerate compression. Unlike linear scaling, which requires hard clipping at extreme values, the sigmoid provides smooth asymptotic saturation at the tails. This structural bound ensures that extreme predictive entropy outliers—such as unconstrained hallucinations or rigidly memorized literal states—gracefully saturate at the operational limits, guaranteeing the system never enters a degenerate compression state where zero or all tokens are retained.

Crucially, beyond budgeting, $H(c)$ explicitly measures the reliability of the few-shot inference; as detailed in Section 3.5, high predictive entropy actively regulates the fusion trust weight α to suppress uncertain LLM hallucinations. This α is heart of the neural fusion.

3.3.3 Selective Token Retention (Local Guardian) Module and Pruned CoT

Generation probability only measures how predictable a token is during LLM generation. It cannot determine if a token actually signals irony. Therefore, we replace probability-based scores with a task-aware piecewise-constant function to isolate the critical tokekns of the CoT. Let $\mathcal{G}_k$ denote the k tokens with highest gradient norms, following gradient-based saliency approaches (Simonyan et al., 2013; Li et al., 2016), computed via backward pass through the RoBERTa classification head:

$$S(c_i) = \|\nabla_{e_{c_i}} \mathcal{L}_{\text{CE}}(\mathbf{z}, y = 1)\|_2, \tag{3}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss between the model logits $\mathbf{z}$ and the ironic target class ($y = 1$), and e_{c_i} is the i-th token embedding. For the entirety of this study, we have considered $k = 5$. This parameter is completely adjustable.

The gradient norm in Equation (3) measures the target-specific sensitivity of the RoBERTa loss to the corresponding token embedding. Mathematically it is a target-specific gradient sensitivity of the loss with respect to the token embedding. It acts as a sensitivity dial: if a specific word drastically spikes the model's irony prediction, it receives a high score, therefore enabling us to rank the tokens. This gradient-based mechanism requires access to the parameters and token embeddings of the downstream RoBERTa classifier, and therefore is not directly applicable to purely black-box classifiers or LLM APIs (e.g., GPT-4o (OpenAI, 2024a), Claude 3.5 Sonnet (Anthropic, 2024)) without a locally accessible differentiable model.

Let $\mathcal{C}$ denote contrastive connectors and $\mathcal{W}$ the English stopword list. The robust importance score is:

$$I_{\text{robust}}(c_i) = \begin{cases} \tau_{\text{crit}} & \text{if } c_i \in \mathcal{G}_k \\ \tau_{\text{conn}} & \text{if } c_i \in \mathcal{C} \\ \tau_{\text{cont}} & \text{if } c_i \notin \mathcal{W},\ |c_i| \geq 3 \\ \tau_{\text{low}} & \text{otherwise.} \end{cases} \tag{4}$$

In this work, we have considered $(\tau_{\text{crit}}, \tau_{\text{conn}}, \tau_{\text{cont}}, \tau_{\text{low}}) = (1000, 500, 10, 1)$. The values are chosen to draw clear boundaries between two adjacent classes of tokens. The multi-order-of-magnitude gaps ensure that gradient-critical tokens and contrastive connectors receive lexicographic retention priority whenever the compression budget admits them. The compressed trajectory is:

$$\tilde{c}_{\text{RDS}} = \text{top}_{\substack{S\subseteq c \\ |S|=\lfloor(1-\gamma_{\text{dyn}})m\rfloor}} I_{\text{robust}}(c_i) \quad \text{(order preserved).} \tag{5}$$

Because $I_{\text{robust}}(c_i)$ can take only four possible values, there can be cases in which some tokens with the same importance score have to be retained while others have to be pruned. As a tie-breaking rule, the tokens are retained sequentially until the retention budget is exhausted.

3.4 Linguistically-Motivated Prior Module

To capture explicit authorial markers that neural pipelines smooth away, we define a symbolic prior

$S_{\text{prior}}(x)$ over tiered signal sets. Let $\mathcal{R}_{\text{strong}}$ denote definitive author-labeled irony markers, $\mathcal{R}_{\text{weak}}$ soft hashtag signals, $\mathcal{R}_{\text{emoji}}$ contradictory emoji patterns, and $\mathcal{R}_{\text{elong}}$ character-elongation tokens. The full lexicon definitions appear in appendix A. The prior is:

$$S_{\text{prior}}(x) = \begin{cases} \rho_{\text{strong}} & \text{if } x \cap \mathcal{R}_{\text{strong}} \neq \emptyset \\ S_{\text{comp}}(x) & \text{otherwise,} \end{cases} \tag{6}$$

where the composite weak score is capped to prevent over-firing:

$$S_{\text{comp}}(x) = \min\Big(\rho_{\text{cap}},\ \delta_w \rho_{\text{weak}} + \delta_e \rho_{\text{emoji}} + \delta_\ell \rho_{\text{elong}}\Big), \tag{7}$$

where δ_w, δ_e, and $\delta_l \in \{0, 1\}$ are binary indicator functions denoting the presence of weak irony hashtags, ironic emojis, and elongated words, respectively. Calibrated values are in Appendix F.5.

3.5 Entropy-Gated Adaptive Inference Fusion

The multi-stage fusion engine balances trust among P_{CoT}, P_{RoBERTa}, and S_{prior}. Crucially, we utilize the predictive entropy $H(c)$ to explicitly measure the confidence of the LLM's inference, allowing the system to penalize the LLM's influence when high entropy indicates internal uncertainty.

Phase 1: CoT Skepticism Gate. To prevent overconfident LLM hallucinations, extreme P_{CoT} predictions are contracted toward 0.5 when $H(c) > H_{\text{sk}}$:

$$\begin{aligned} &\text{If } H(c) > H_{\text{sk}} \text{ and } |P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) - 0.5| \geq 0.5 - p_{\text{sk}}: \\ &\quad \tilde{P}_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) = 0.5 \\ &\qquad + \lambda_{\text{sk}}\big[P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) - 0.5\big]; \\ &\text{else} \\ &\quad \tilde{P}_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) = P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}). \end{aligned} \tag{8}$$

Phase 2: Base Fusion. The trust weight α (confidence in the RoBERTa baseline) is linearly interpolated over $[H_{\text{min}}, H_{\text{max}}]$ with bounds $[\alpha_{\text{min}}, \alpha_{\text{max}}]$. The base fusion is then computed as:

$$P_{\text{base}}(\hat{y} = 1|x) = \alpha P_{\text{RoBERTa}}(\hat{y} = 1|x) + (1 - \alpha)\tilde{P}_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}). \tag{9}$$

The calibrated hyperparameter values for the gating and fusion parameters are provided in Appendix F.5.

Phase 3: Prior Injection. The final probability interpolates the base fusion with the symbolic prior:

$$P(\hat{y} = 1|x) = \omega P_{\text{base}}(\hat{y} = 1|x) + (1-\omega) S_{\text{prior}}(x). \tag{10}$$

For strong signals, α is first capped at $\alpha_{\text{cap}} = 0.30$, and a heavy blend is applied ($\omega = \omega_{\text{strong}} = 0.60$). For weak signals, a moderate blend is applied ($\omega = \omega_{\text{weak}} = 0.75$). If no signal is present, the system defaults to $P(\hat{y} = 1|x) = P_{\text{base}}(\hat{y} = 1|x)$. The final decision is $\hat{y} = 1$ if $P(\hat{y} = 1|x) > 0.5$.

4 Experimental Setup

4.1 Datasets

TweetEval Irony (SemEval-2018 Task 3 (Van Hee et al., 2018)) : 784 annotated tweets serving as primary evaluation benchmark. We designate the first 50 tweets as a validation split for parameter calibration; all reported results are on the strictly held-out 734-tweet test set. For our experiments, we use the TweetEval Irony dataset exactly as distributed in its official GitHub repository, using the provided tweet texts without modification.

iSarcasm (SemEval-2022 Task 6, English Task A) : 1400 self-reported sarcasm tweets ($N_{\text{ironic}} = 200$, $N_{\text{literal}} = 1200$). The severe class imbalance and implicit-intent distribution provide an out-of-distribution stress test. The RDS Fusion framework is deployed *without any parameter modification* on this dataset.

4.2 Models and Evaluation Pipeline

Three components constitute the pipeline: (1) `Qwen2.5-3B-Instruct` generates CoT trajectories under 4-bit NF4 quantization, greedy decoding, `max_new_tokens=120`; (2) the **Local Guardian** classifies the tokens into four classes based on importance; (3) a post-generation compression procedure applies the resulting importance scores to construct the compressed CoT, while **cardiffnlp/twitter-roberta-base-irony** provides the discriminative baseline and gradient signals. Hardware and library details are in appendix F.

4.3 Evaluated Configurations

Three configurations isolate each component's contribution: (i) **RoBERTa-only baseline**: task-specific fine-tuned encoder (frozen); (ii) **Ablation** (RoBERTa + Hashtag Prior): symbolic prior in-

jected without CoT; (iii) **RDS Fusion**: complete neuro-symbolic pipeline.

A standalone few-shot LLM configuration was evaluated but excluded from the primary tables due to a very weak performance (Macro F1 < 0.500), validating its restricted role as a supplementary filter.

5 Results and Discussion

5.1 Necessity of Post-Generation Pruning

A central component of RDS Fusion is the Local Guardian, which performs post-generation pruning of the LLM reasoning trajectory before it is passed to the fusion layer. To examine whether this pruning is necessary, we compare the complete RDS Fusion framework against an otherwise identical configuration, denoted as RDS Fusion (no pruning), in which the generated CoT is passed to the fusion layer without rule-based token pruning. Both configurations use the same LLM, generation budget, symbolic prior, RoBERTa classifier, and adaptive fusion mechanism; the only difference is the application of the Local Guardian.

The results demonstrate the necessity of the pruning: RDS Fusion achieves an accuracy of **78.1%**, compared with **75.5%** for RDS Fusion (no pruning), yielding an absolute improvement of **2.6 percentage points**. Since the two configurations differ only in whether the Local Guardian is applied, the improvement indicates that the post-generation pruning is necessary. In this sense, it is not merely a mechanism for reducing the representation size of CoT; it acts as a task-aware filtering mechanism that improves the quality of the reasoning signal supplied to the final fusion stage.

The slightly higher ironic-class F1 of RDS Fusion (no pruning) (0.759 vs. 0.747) indicates that retaining the complete generated CoT can provide a marginal advantage for identifying ironic instances. However, this comes at the cost of lower overall performance, with RDS Fusion achieving substantially higher accuracy and Macro F1. Thus, pruning appears to trade a small amount of ironic-class F1 for a better-balanced overall classifier. All the numbers are given in table 1. It is to be noted that the run without token pruning costs marginally lower time than the one with post-generation pruning.

5.2 System-Level Performance

In RDS Fusion, LLM functions as a precision-restoration filter—rescuing 68 false positives (FP: $178 \rightarrow 110$) while maintaining ironic recall of 0.824. This calibration yields 78.1% accuracy and Macro F1 of 0.777.

The structural necessity of the Chain-of-Thought pipeline is visually confirmed across system metrics. As illustrated in Figure 2 and the corresponding confusion matrices (Figure 4), the symbolic ablation artificially maximizes ironic recall at the cost of a severe precision collapse, generating 178 false positives. RDS Fusion dynamically suppresses these heuristics to recover precision, systematically reducing false positives to 110. Furthermore, this precision recovery is tightly calibrated to reasoning complexity; Figure 3 demonstrates that accuracy degrades logically as predictive entropy increases, whereas the ablation's error distribution remains entirely uncorrelated with predictive ambiguity.

Table 1 presents the three-way comparison on the held-out TweetEval test set. The standalone RoBERTa achieves 73.2% accuracy with a low false positive rate but suffers from ironic recall of only 0.557. Injecting the symbolic prior (ablation) raises accuracy to 75.7% but at a critical cost: the fusion formula forces ironic recall to 1.000 (0 false negatives) because the symbolic prior activates on all gold-ironic instances in this partition, generating 178 false positives and collapsing precision to 0.619. The Macro F1 of 0.757 therefore does not fully reflect the severe precision collapse caused by the recall-saturated operating point.

5.3 Statistical Significance

Paired McNemar's tests across configurations were conducted to evaluate statistical significance. It was found that, adding the symbolic prior alone over the RoBERTa baseline does not yield significance ($\chi^2 = 1.37$, $p = 0.242$)—the recall gains are neutralized by the 178 false positives. The direct comparison between RDS Fusion and ablation also falls short of significance ($\chi^2 = 2.08$, $p = 0.149$). Only the RDS Fusion Fusion framework achieves a statistically validated improvement over the RoBERTa baseline ($\chi^2 = 7.95$, $p = 0.005$).

5.4 Subpopulation Analysis

Partitioning the test set by symbolic-prior activation provides a targeted analysis of the interaction between the heuristic module and the CoT-based fusion mechanism. The prior is activated on 417 tweets, while 317 tweets receive no symbolic-prior signal. On the prior-activated subset, the RoBERTa + Prior ablation achieves an Ironic F1 of 0.829 but a

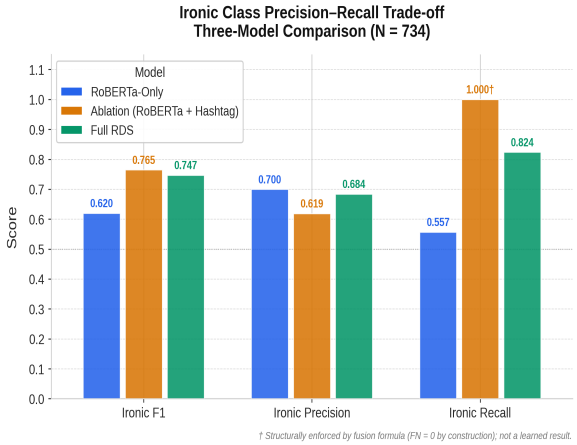


Figure 2: Ironic class metrics across configurations. Ablation's perfect recall is a structural artifact forcing precision collapse; CoT restores balance.

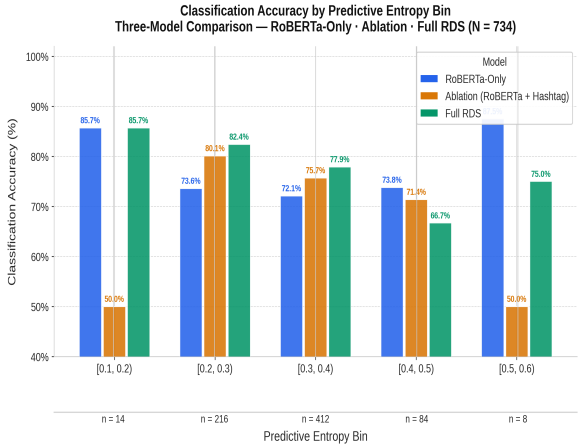


Figure 3: Accuracy by entropy bin ($N = 734$). Accuracy degrades monotonically with ambiguity; the ablation is uncorrelated with entropy.

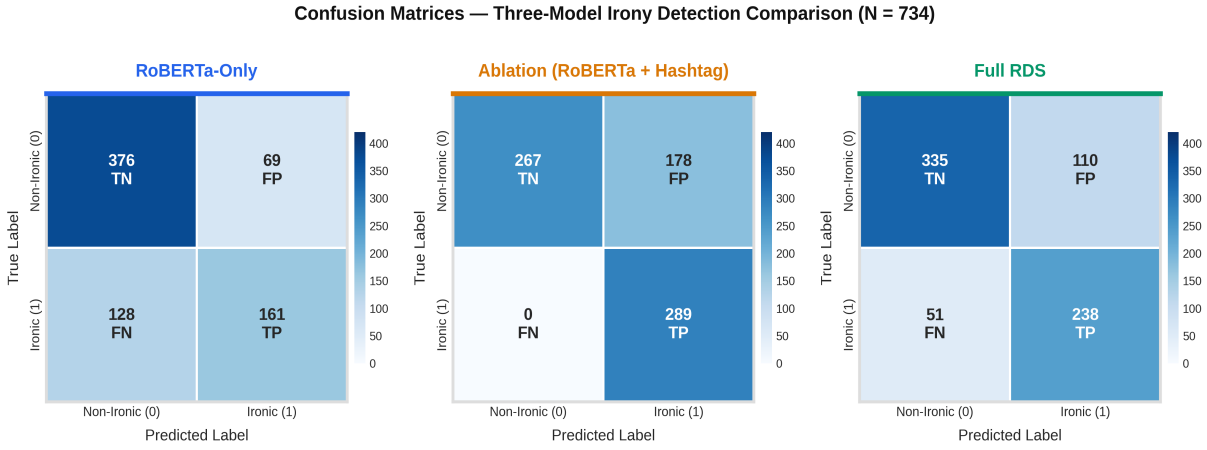


Figure 4: Confusion matrices for all three TweetEval configurations. The RoBERTa baseline is conservative (FP=69, FN=127); the ablation enforces zero false negatives at the cost of FP=178; RDS Fusion fusion reduces FP to 110 while keeping FN=51.

severely degraded Non-Ironic F1 of 0.131, reflecting the tendency of explicit markers to over-fire on literal tweets. RDS Fusion raises Non-Ironic F1 from 0.131 to 0.508 while maintaining comparable accuracy (71.5% vs. 71.7%) and a high Ironic F1 of 0.801. This provides direct evidence that the CoT signal acts as a semantic safeguard against false positives introduced by the symbolic prior.

We found that The non-triggered subset contains **no gold-ironic example** in this evaluation partition. Consequently, its Ironic F1 is undefined. This reflects the absence of positive instances in this subset, rather than a model classification outcome.

5.5 Calibration Dynamics

We observe a statistically significant negative correlation between entropy and accuracy for RDS ($r = -0.104$, $p = 0.005$), versus no correlation for the ablation ($r = -0.028$, $p = 0.456$). The 10.9 percentage-point accuracy spread across entropy quartiles (vs. 3.9 for the ablation) is a direct evidence that entropy actively quantifies structural ambiguity rather than merely measuring text difficulty. The α distribution clusters heavily at the $[0.30, 0.40]$ floor (55.4% of inferences), confirming the system's conservative failsafe design. However, $\alpha > 0.5$ on 41.3% of inferences demonstrates the parameter's active contribution.

While γ_{dyn} was introduced to provide an inference-time analogue of TokenSkip's adaptive compression behavior, we report a negative result regarding its macro-level efficacy. Unlike mathematical reasoning, where generation lengths vary drastically between complex proofs and sim-

Method	Accuracy (%)	Macro F1	Ironic F1	Ironic Precision	Ironic Recall	Non-Ironic F1
RoBERTa-only	73.2	0.706	0.620	0.700	0.557	0.792
Ablation (RoBERTa + Prior)	75.7	0.757	0.765	0.619	1.000†	0.750
Unrestricted CoT Ablation	74.9	0.749	0.755	0.614	0.979	0.743
RDS Fusion (no pruning)	75.5	0.755	0.759	0.619	0.979	0.751
RDS Fusion	**78.1**	**0.777**	**0.747**	**0.684**	**0.824**	**0.806**

Table 1: Performance on TweetEval Irony test set ($N = 734$). †Ablation's ironic recall of 1.000 is a structural artifact forcing precision collapse.

Subset	Model	Accuracy (%)	Macro F1	Ironic F1	Non-Ironic F1
Triggered ($n = 417$)	Ablation (RoBERTa + Prior)	71.5	0.480	0.829	0.131
	RDS Fusion	**71.7**	**0.655**	**0.801**	**0.508**
Non-Triggered ($n = 317$)	Ablation (RoBERTa + Prior)	81.4	0.449	—	0.897
	RDS Fusion	**86.4**	**0.464**	—	**0.927**

Table 2: Full subpopulation analysis based on symbolic prior activation. CoT precision recovery elevates Non-Ironic F1 from 0.131 to 0.508 on triggered tweets.

ple calculations, our prediction task yields a comparatively constrained CoT length (mean=83.7 tokens, $\sigma = 5.39$, range 60–104). Consequently, γ_{dyn} exhibits near-zero variance (mean = 0.466, $\sigma = 0.013$). However, the attribution of this negative result to the uniform nature of the task has not been validated by us. This presents a room for future work.

5.6 Comparison with Unrestricted CoT

To compare RDS Fusion with unconstrained reasoning, we conducted an unrestricted CoT ablation using the same LLM and downstream classifier, but without post-generation CoT compression and with a substantially larger generation budget of 1000 tokens. On the complete 734-example evaluation set, unrestricted CoT achieved 74.9% accuracy and an ironic-class F1 of 0.755, compared with 78.1% accuracy and 0.747 ironic F1 for RDS Fusion. While unrestricted reasoning therefore achieves a marginally higher ironic-class F1, it does so with a markedly different precision–recall profile: its ironic precision is only 0.614, compared with 0.684 for RDS Fusion, while its recall reaches 0.979 compared with 0.824 for RDS Fusion. Thus, increasing the reasoning budget substantially improves recall but does not recover the precision lost relative to the RoBERTa baseline. In contrast, RDS Fusion provides a more balanced precision–recall trade-off while achieving higher overall accuracy and Macro F1 (0.777 vs. 0.749). Furthermore, RDS Fusion was observed to be more than **2.5 times faster** than the unrestricted configuration. These results indicate that substantially longer reasoning is neither necessary nor sufficient for improved overall performance; the results of the unrestricted CoT ablation are discussed in detail in appendix B.5.

5.7 Comparison with Existing Supervised Frameworks

Table 3 shows RDS Fusion achieving comparability with BERTweet-base (Nguyen et al., 2020) in overall accuracy (78.1% vs. 78.2%) while slightly outperforming it on Ironic F1 (0.747 vs. 0.746). Furthermore, RDS Fusion achieves a dominant Macro F1 score of 0.777, substantially surpassing CCR-Net's (Rahman et al., 2026) 0.708 and demonstrating balanced predictive performance across both the ironic and non-ironic classes—entirely without task-specific fine-tuning. However, the 50-tweet calibration holdout prevents a strictly identical numerical comparison, but this performance strongly suggests that well-calibrated neuro-symbolic fusion can match or exceed the output of domain-trained encoders.

Model	Paradigm	Acc.	MF1	Ir.F1
CCR-Net	Supervised	70.9	0.708	0.685
BERTweet-base	Supervised	**78.2**	—	0.746
RDS Fusion	Hybrid	78.1	**0.777**	**0.747**

Table 3: Comparison with published supervised systems on the TweetEval Irony task. BERTweet-base and CCR-Net results are reported on the full 784-tweet test split; RDS Fusion is evaluated on a held-out subset of $N = 734$.

5.8 Few-Shot Cross-Domain Robustness on iSarcasm

Deployed without modification on iSarcasm, the RoBERTa baseline generates 315 false positives

due to distribution shift. The RDS Fusion framework suppresses 22.5% of these false positives ($315 \rightarrow 244$; McNemar's $\chi^2 = 21.44$, $p < 0.0001$), elevating Macro F1 from 0.6562 to 0.6726. Crucially, this precision recovery does not degrade recall; RDS Fusion marginally elevates the baseline's Ironic F1 ($0.4786 \rightarrow 0.4821$).

Model	Eval.	Ir.F1
stce (Yuan et al., 2022)	Supervised SOTA	0.6052
RDS Fusion	**Hybrid**	**0.4821**
Ablation (RoBERTa + Symbolic Prior)	Hybrid	0.4793
RoBERTa-only Baseline	Supervised†	0.4786
SarcasmDet (Abdullah et al., 2022)	Supervised	0.4300
UTNLP (Abaskohi et al., 2022)	Supervised	0.3800
FII UAIC (Manoleasa et al., 2022)	Supervised	0.3700

Table 4: SemEval-2022 Task 6 leaderboard (selected subset) vs. RoBERTa, the Ablation Model & RDS Fusion. †RoBERTa is trained on the train set of TweetEval training set.

As shown in table 4, few-shot RDS Fusion outperforms three out of the four selected, supervised SemEval submissions. However, iSarcasm also exposes architectural limits: the symbolic prior fires on only 1.5% of samples ($n = 21$), confirming its inertia on implicit intent. On the 21 triggered instances, frozen TweetEval weights allow the LLM's literal confidence to overpower valid priors (Ironic F1: $0.889 \rightarrow 0.571$), defining the cross-domain calibration boundary.

Despite this localized symbolic collapse, the framework's overall outperformance of in-domain SemEval models isolates the value of the generative CoT filter. By successfully mitigating the baseline's out-of-distribution hallucinations without requiring any iSarcasm-specific fine-tuning, RDS Fusion shows that post-hoc generative reasoning can effectively substitute for domain-specific retraining when tackling novel datasets.

5.9 A Case Study: Precision Recovery on a Literal Legal Headline (Tweet 86, TweetEval Dataset)

Tweet: *#Myanmar #men #plead #not #guilty to #murder of #British #tourists...*

This is a literal news headline. The author has tokenized the text using individual hashtags, including the isolated substring `#not`. The explicit symbolic prior immediately identifies this as a strong irony signal ($S_{\text{HTag}} = 0.88$).

Ablation failure: Despite RoBERTa's low baseline probability ($P_{\text{RoBERTa}}(\hat{y} = 1|x) = 0.209$), the strong symbolic trigger pushes the fused probability above 0.500, yielding a false positive. This is the canonical structural artifact of the over-aggressive prior.

RDS Fusion correction: The CoT pipeline parses the semantic continuity of the legal phrase "plead not guilty" rather than treating "#not" as an isolated pragmatic inversion. During compression, the Local Guardian explicitly whitelists literal anchoring tokens (*British*, *indicates*) against the budgeter. The compressed CoT gives $P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) = 0.15$. Routed at $\alpha = 0.300$, this contradicts the prior and pulls the fused probability to $P(\hat{y} = 1|x) = 0.453$, yielding a correct Non-Ironic prediction. This interaction demonstrates exactly how the CoT engine functions as a semantic safeguard against symbolic over-firing.

Two more case studies are provided in appendix D.

6 Conclusion

We presented Robust Dual-Signal (RDS) Fusion, a training-free-at-inference hybrid neuro-symbolic inference pipeline achieving few-shot irony detection at a performance level comparable to supervised systems. On the held-out TweetEval test ($N = 734$), RDS Fusion achieves numerical comparability with BERTweet-base and significantly improves upon the frozen, fine-tuned RoBERTa baseline ($p = 0.005$). Without modification, it also surpasses multiple supervised SemEval systems on the cross-domain iSarcasm benchmark. Our findings suggest that unrestricted reasoning is not always better for a prediction task, specifically when a compact LLM is used. We also found that, the compressed (pruned) CoT in RDS Fusion functions as a *precision-recovery* engine. However, neither the unrestricted CoT ablation and nor the ablation with no post-generation pruning showed this behaviour—this was solely observed for the constrained and post-pruned CoT component. These results show that calibrated frameworks requiring no additional task-specific fine-tuning of the LLM or fusion components can match or even exceed the performance of supervised frameworks.

7 Limitations

Inference Latency. The framework incurs a multi-order-of-magnitude latency penalty compared to the baseline: the RoBERTa baseline processes 1400 samples in <1 minute; RDS Fusion requires ≈5 hours (≈12.86 sec/sample) due to sequential autoregressive generation by the LLM and per-sample backward passes. RDS Fusion is appropriate only in cases where supervised finetuning is not feasible due to time or computational costs. However, the framework is still a number of times faster than one that utilizes the whole, unrestricted and uncompressed CoT.

LLM Inference Variability. The LLM inference process is not guaranteed to produce an identical CoT signal for the same tweet across separate runs, making the resulting reasoning signal potentially run-dependent. Consequently, the CoT-derived component may vary across repeated evaluations even when the input tweet remains unchanged. Therefore, the overall result also may demonstrate small variations across different runs.

Nearly Static Dynamic Compression Ratio. We introduced the dynamic compression ratio so that tokens could be pruned depending on the particular CoT. However, it was found that γ_{dyn} was almost always in the range of $0.45 - 0.51$. The near-zero variance of the dynamic retention budget (γ_{dyn}) is a negative result driven likely by the uniform nature of the task of prediction. The semantic limitations of the compact 3B-parameter model may also have compounded this inertia, however, it remains an area of future validation.

Limitation of the Symbolic Prior. On datasets where irony is primarily psychological or semantic rather than syntactic (e.g., iSarcasm), the symbolic prior remains largely dormant, activating on only very few of the samples (1.5% in case of iSarcasm). In cases where there are literally no explicit symbolic signals, RDS Fusion becomes a purely neural cofiguration.

Static Fusion Weight Fragility. Calibration thresholds optimized on a 50-tweet TweetEval split cannot universally transfer across shifting linguistic distributions. On iSarcasm's triggered subset, frozen blending weights allowed the LLM's literal bias to overpower the valid symbolic prior, degrading Ironic F1 from 0.889 to 0.571.

Acknowledgements

We sincerely thank Professor Sourangshu Bhattacharya for his guidance as the instructor of CS60050, where the idea of RDS Fusion originated as a course project. We also thank Vaishnovi Arun for mentoring our project group, and Yuvraj Veer and Maitreyee Chakraborty for their contributions during the project. We are grateful to Professor Rajaram Lakkaraju and Professor Niloy Ganguly for their support following the acceptance of this work.

Ethics Statement

This work uses publicly available social media datasets, `TweetEval` and `iSarcasm`, for research on automatic irony detection. We do not collect new user data or conduct experiments involving human participants. The datasets may contain offensive or sensitive language inherent to social media. The complete RDS Fusion implementation and generation logs are available in its `GitHub` repository, and the `twitter-roberta-base-irony` model is publicly available via Hugging Face.

References

Amirhossein Abaskohi, Arash Rasouli, Tanin Zeraati, and Behnam Bahrak. 2022. UTNLP at SemEval-2022 task 6: A comparative analysis of sarcasm detection using generative-based and mutation-based data augmentation. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 962–969.

Malak Abdullah, Dalya Alnore, Safa Swedat, Jumana Khrais, and Mahmoud Al-Ayyoub. 2022. SarcasmDet at SemEval-2022 task 6: Detecting sarcasm using pre-trained transformers in english and arabic languages. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1025–1030.

Anthropic. 2024. Claude 3.5 sonnet model card addendum. Anthropic.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Artur d'Avila Garcez and Luis C. Lamb. 2023. Neurosymbolic AI: The 3rd wave. *Artificial Intelligence Review*, 56(11):12387–12406.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems*, volume 34.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. 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)*, pages 757–762.

Henry A. Kautz. 2022. The third ai summer: Aaai robert s. engelmore memorial lecture. *AI Magazine*, 43(1):105–125.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. 2023. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6342–6353.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Tudor Manoleasa, Daniela Gifu, and Iustin Sandu. 2022. FII UAIC at SemEval-2022 task 6: isarcasmeval – intended sarcasm detection in english and arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 970–976.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for english tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14.

OpenAI. 2024a. GPT-4o system card. OpenAI.

OpenAI. 2024b. Learning to reason with llms. OpenAI Blog.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. 2024. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981.

Zaryab Rahman, Fakhrud Din, and Shah Khalid. 2026. CCR-Net: Surpassing baselines in irony detection via supportive-conflictive attention fusion. *Expert Systems with Applications*, 304:130814.

Antonio Reyes, Paolo Rosso, and Tony Veale. 2013. A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation*, 47(1):239–268.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Preprint*, arXiv:1312.6034.

Cynthia Van Hee, Els Lefever, and Veronique Hoste. 2018. SemEval-2018 task 3: Irony detection in english tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, pages 39–50.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Heming Xia, Yongqi Li, Chak Tou Leong, Wenjie Wang, and Wenjie Li. 2025. Tokenskip: Controllable chain-of-thought compression in llms. *Preprint*, arXiv:2502.12067.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2024. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Mengfei Yuan, Mengyuan Zhou, Lianxin Jiang, Yang Mo, and Xiaofeng Shi. 2022. stce at SemEval-2022 task 6: Sarcasm detection in english tweets. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 820–826.

A Extended Neuro-Symbolic Heuristics & Lexicons

A.1 Contrastive Connector Vocabulary ($\mathcal{C}$)

The contrastive connector set used in the Local Guardian is:

$\mathcal{C} = \{$*but, yet, though, although, however, despite, whereas, still, even, never, not, no, without, barely*$\}$.

These tokens are disproportionately responsible for carrying ironic inversion in natural language (Joshi et al., 2015; Reyes et al., 2013) and receive lexicographic retention priority for any compression budget, by virtue of $\tau_{\text{conn}} = 500 \gg \tau_{\text{cont}} = 10$.

A.2 Symbolic Prior Lexicons

Strong signal set $\mathcal{R}_{\text{strong}}$: definitive author-labeled irony hashtags and their close orthographic variants:

```
#sarcasm, #sarcastic, #irony,
#ironic, #not, #justkidding, #jk
```

Presence of any element assigns $S_{\text{prior}}(x) = \rho_{\text{strong}} = 0.88$.

Weak signal set $\mathcal{R}_{\text{weak}}$: soft hashtag signals often co-occurring with ironic intent but insufficient alone:

```
#lol, #sure, #totally,
#obviously, #right, #yeah,
#great, #awesome, #fantastic,
#wonderful, #perfect, #love,
#amazing
```

Each contributes $\rho_{\text{weak}} = 0.15$ to the composite score.

Emoji contrast rules $\mathcal{R}_{\text{emoji}}$: structural patterns where a positive-valence emoji is immediately followed by a negative-valence emoji or vice versa within the same token window. Positive cluster includes smiling, laughing, heart, thumbs-up, and celebration symbols. Negative cluster includes crying, angry, broken-heart, and eye-roll symbols. A match contributes $\rho_{\text{emoji}} = 0.25$.

Elongation tokens $\mathcal{R}_{\text{elong}}$: tokens where any single character is consecutively repeated $\geq \eta = 3$ times (e.g., *sooooo, yesssss, reallyyyy*). Detected via the regex `(.)\1{2,}`. Contributes $\rho_{\text{elong}} = 0.12$.

A.3 Negation-Aware Semantic Scan Vocabulary

Ironic indicator tokens :
`sarcasm, sarcastic, ironic, irony, clearly, obviously, totally, definitely, absolutely, sure, right, great, wonderful, perfect, love, exactly, certainly, naturally.`

Literal indicator tokens :
`literal, sincere, genuine, actually, truly, honestly, factual, real, straightforward, serious, earnest, authentic.`

Local negation tokens (invert the score of the subsequent indicator):
`not, no, never, without, hardly, barely, cannot, can't, won't, don't.`

B Chain-of-Thought Generation and Compression

B.1 Few-Shot Chain-of-Thought Prompt Template

The prompt fed to the LLM for the chain-of-thought generation is as follows:

```
You are an expert linguist specializing in detecting irony and sarcasm in social media text.

DEFINITION: A tweet is IRONIC if there is a contrast between its literal meaning and its intended meaning, or if the author says the opposite of what they actually mean (often to mock, criticize, or be humorous). A tweet is NON-IRONIC if it is a sincere, literal statement.

KEY SIGNALS TO CHECK: - Does the literal meaning contradict the real-world situation? - Is there an exaggerated, over-the-top positive/negative tone? - Are there hashtags like #not, #sarcasm, #irony, #obviously that signal ironic intent? - Does the tweet mock or criticize something by pretending to praise it? - Would a reasonable reader take this at face value, or detect a hidden meaning? - Are there elongated words like "Loooove" or "Soooo" used sarcastically?

CONFIDENCE SCALE: 0.0 = Absolutely certain NON-IRONIC
```

(sincere, literal, no ambiguity at all); 0.1 = Very likely non-ironic, tiny residual doubt; 0.3 = Probably non-ironic, some mixed signals present; 0.5 = Completely uncertain — could genuinely be either; 0.7 = Probably ironic, some mixed signals present; 0.9 = Very likely ironic, tiny residual doubt; 1.0 = Absolutely certain IRONIC (clear sarcasm/irony, no ambiguity at all).

EXAMPLES:

Tweet: "Oh great, another Monday. Just what I needed."
Reasoning: "Just what I needed" is exaggeratedly positive about something universally disliked. Classic sarcasm with no ambiguity. Score: 0.95.

Tweet: "Happy birthday to my best friend! Hope your day is amazing."
Reasoning: Sincere, literal birthday wish. No hidden meaning, no contrast, tone matches content perfectly. Score: 0.05.

Tweet: "Wow, love how my flight got cancelled on the day of my interview. Truly blessed."
Reasoning: "Truly blessed" after describing a disaster is a clear ironic inversion. Very high confidence. Score: 0.92.

Tweet: "This weather is something else today."
Reasoning: Ambiguous — could be genuine admiration or sarcastic complaint depending on context not available in the tweet alone. Score: 0.50.

Now analyze the following tweet using the same reasoning process.

Tweet: {tweet}.

Think step by step through the KEY SIGNALS above. End your response with EXACTLY: **Score:** X.XX (a number between 0.00 and 1.00, two decimal places).

B.2 Role of Compression and Difference with TokenSkip

Role of compression. Our framework employs a two-stage compression process. First, the LLM generation is bounded by a fixed budget of `max_new_tokens=120`, which can truncate reasoning before the LLM reaches its complete inference or explicit confidence score. Second, a post-generation pruning mechanism selectively retains the most downstream-relevant tokens from the generated sequence. Thus, the compressed CoT is constructed from whatever reasoning is available within the generation budget, rather than necessarily from the LLM's complete reasoning trajectory. When the LLM's explicit numerical score is pruned or if it is not available at all within the original reasoning length, we compute a new CoT-based confidence estimate from the surviving reasoning. Consequently, the resulting P_{CoT} is a compression-derived estimator and need not coincide with the LLM's original confidence. This distinction, and the motivation for discarding the original score, are discussed in in the following subsection.

Difference from TokenSkip. Here, the second stage of compression is performed through **post-generation token-pruning**, whereas in frameworks like TokenSkip, compression is learned during supervised fine-tuning and performed directly during autoregressive generation.

B.3 Motivation for Discarding Pruned Confidence Scores

It was observed that the restricted CoT rarely ended with a numerical score, Therefore in most cases, scores had to be computed via the methods described in subsection 3.3.1. However, even if the score was computed, that was often discarded, as it was rarely classified as critical by the gradient-based scoring module.

When the token encoding the LLM's explicit confidence score is pruned, its numerical prediction is discarded rather than retained as an independent signal. This design follows directly from the gradient-based definition of token relevance. For a token embedding e_{c_i}, a first-order perturbation gives

$$\Delta\mathcal{L}_{\mathrm{CE}} \approx (\nabla_{e_{c_i}}\mathcal{L}_{\mathrm{CE}})^{\top}\Delta e_{c_i}. \tag{11}$$

Thus, a token with low gradient sensitivity has limited local influence on the downstream classification objective. Retaining its explicit numerical score after pruning would allow information associated with a token deemed task-irrelevant to bypass the compression mechanism. We therefore reconstruct P_{CoT} from the surviving reasoning, maintaining consistency between token selection and the resulting CoT-derived signal.

This design choice does not imply that the raw LLM score is intrinsically uninformative. Rather, it ensures that the compressed CoT signal is derived exclusively from the evidence retained by the task-aware selector.

B.4 Two-Stage CoT Compression and Confidence Reconstruction

RDS Fusion applies CoT compression in two stages:

1. **Generation-time token constraint.** `Qwen2.5-3B-Instruct` generates the reasoning trajectory with `max_new_tokens=120`. This is a hard upper bound, not a target length, and therefore limits autoregressive generation cost. A trajectory may consequently terminate before the model completes its reasoning or produces an explicit confidence score.

2. **Post-generation pruning.** The generated trajectory is subsequently processed by the Local Guardian. Tokens with insufficient downstream sensitivity are removed while preserving the order of retained tokens, producing the compressed trajectory $\tilde{c}$ used by the fusion layer.

These two stages have different computational roles: the generation-time constraint bounds autoregressive computation, whereas post-generation pruning reduces the representation passed to downstream components. The latter cannot retroactively reduce the cost of tokens that have already been generated.

We distinguish the LLM's original confidence from the confidence used by RDS. Let P_{LLM} denote an explicit numerical confidence expressed in an uncompressed LLM trajectory, when available, and let P_{CoT} denote the estimator extracted from the compressed trajectory $\tilde{c}$. In general,

$$P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}}) \neq P_{\mathrm{LLM}}(\hat{y}=1|c).$$

Formally, given the compressed trajectory $\tilde{c}_{\mathrm{RDS}}$, the following steps are followed sequentially to obtain $P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}})$:

1. **Explicit score:** search $\tilde{c}_{\mathrm{RDS}}$ for `Score:` s, and set
$$P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}}) = \mathrm{clip}(s, 0, 1).$$

2. **Verdict fallback:** if no score exists, parse a `Verdict:` statement. Set $P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}}) = 0.9$ for an ironic/sarcastic verdict and $P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}}) = 0.1$ for a non-ironic/literal verdict.

3. **Semantic fallback:** if neither is found, scan the final three sentences of $\tilde{c}_{\mathrm{RDS}}$. Let r_i and r_l denote the weighted counts of ironic and literal indicators, respectively. Then
$$P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}}) = 0.15 + \frac{0.7 r_i}{r_i + r_l},$$
clipped to $[0.15, 0.85]$.

4. **Full-text fallback:** if the semantic scan yields no indicators, apply the same mapping to the full $\tilde{c}_{\mathrm{RDS}}$.

5. **Forced verdict:** if $P_{\mathrm{CoT}}(\hat{y}=1|\tilde{c}_{\mathrm{RDS}}) = 0.5$, reprompt the LLM using the compressed trajectory and a 10-token forced-verdict template, then repeat Steps 1–4.

Computation of Semantic Indicator Scores. Let $\mathcal{I}$ and $\mathcal{L}$ denote the ironic and literal indicator vocabularies defined in Appendix A.3. For each occurrence w of an indicator in the selected CoT span, we assign a context-sensitive weight

$$V(w) = \begin{cases} 3, & \text{for a contextual conclusion cue,} \\ 2, & \text{otherwise.} \end{cases}$$

Thus, each indicator occurrence receives a context-sensitive weight: $V(w) = 3$ in a concluding context and $V(w) = 2$ otherwise. The weighting is applied independently to each indicator; hence, "*clearly ironic*" contributes $3 + 3 = 6$ when both indicators occur in a concluding context. Local negation reverses the polarity, so a negated ironic indicator contributes to r_l, while a negated literal indicator contributes to r_i. The resulting weighted scores are therefore

$$r_i = \sum_{w \in \mathcal{I}} V(w), \qquad r_l = \sum_{w \in \mathcal{L}} V(w).$$

If $r_i = r_l = 0$, no semantic indicator is present and the pipeline enters into full-text fallback mechanism.

B.5 Unrestricted Reasoning Ablation

To evaluate whether allowing the LLM to generate substantially longer reasoning improves classification performance, we conducted an unrestricted-CoT ablation using the same `Qwen2.5-3B-Instruct` model and downstream RoBERTa classifier, but without post-generation CoT compression. The generation budget was increased to `max_new_tokens=1000` to allow the reasoning trajectory to evolve to its full extent. This value is nevertheless still a hard upper bound, and trajectories reaching 1000 tokens remain truncated. In practice, the generated unrestricted CoTs averaged approximately 200–500 tokens, substantially longer than the 120-token limit imposed in RDS Fusion before generation. The larger generation budget consequently incurred a considerably higher autoregressive computational cost.

B.5.1 Accuracy and Latency

On the complete 734-example evaluation set, unrestricted CoT with `max_new_tokens=1000` achieved 74.9% accuracy, compared with 78.1% for RDS Fusion under the 120-token constraint. Thus, substantially increasing the reasoning budget did not improve overall performance. RDS Fusion required 2.8 hour to complete evaluation on the whole testset, whereas the ablation took 7.6 hour.

B.5.2 No Recallibration of α

It was also observed that for length-unrestricted reasonings, the entropies were larger than the case when the CoT was restricted to a threshold, i.e. the compressed case. This is expected from the definition of Shannon entropy. However, we did not re-calibrate α for this experiment. It was also observed that for this case, the range of variation entropy was actually quite large. However since the LLM almost always gave one sided / binary predictions (e.g., $0.00, 0.05, 0.92, 0.95$ etc), with estimations of more than 0.8 in most of the cases, re-calibrating to increase α would only have made the performance worse. This is supported by the fact that there were only **8 false negatives** for this ablation.

B.5.3 Precision–Recall Analysis

The precision–recall behavior further illustrates the distinction between the three configurations. Table 5 compares the ironic-class precision and recall of the RoBERTa-only baseline, the unrestricted CoT ablation, and RDS Fusion.

Method	Precision	Recall
RoBERTa-only	0.700	0.557
Unrestricted CoT	0.614	0.979
RDS Fusion	0.684	0.824

Table 5: Ironic-class precision and recall across the baseline, unrestricted CoT ablation, and RDS Fusion.

RoBERTa-only exhibits relatively high precision but substantially lower recall, whereas unrestricted CoT achieves very high recall at the cost of markedly reduced precision. RDS Fusion occupies an intermediate regime, recovering most of the precision of RoBERTa while retaining substantially higher recall than the baseline. Thus, the compressed reasoning signal, when incorporated through the RDS Fusion mechanism, mitigates the strong recall bias observed in unrestricted CoT while avoiding the low-recall behavior of the RoBERTa-only configuration.

B.6 Analysis of the Compressed CoT

Upon analysis of the 5 critical tokens across multiple cases, it was revealed that extracted critical tokens predominantly consist of the LLM's generated reasoning vocabulary (e.g., “contradiction", “mocking"), however tokens from the raw tweets are also present in some cases. This confirms the module functions as a task-aware Chain-of-Thought distiller, isolating the LLM's core pragmatic analysis to feed the fusion layer. Furthermore, it was also found that the original score given by the LLM in its uncompressed CoT, was frequently discarded and through the mechanisms described in subsection 3.3.1, a new score was generated from the compressed CoT.

C Additional Results

C.1 CoT and Adaptive Fusion Diagnostics

Figure 5 shows the distribution of budget-limited CoT lengths **before post-generation pruning**. The lengths are tightly concentrated, with a mean of 83.7 tokens and standard deviation of 5.39, and remain below the 120-token generation limit.

Figure 6 shows the distribution of γ_{dyn}. The small standard deviation ($\sigma = 0.013$) indicates limited variation across instances.

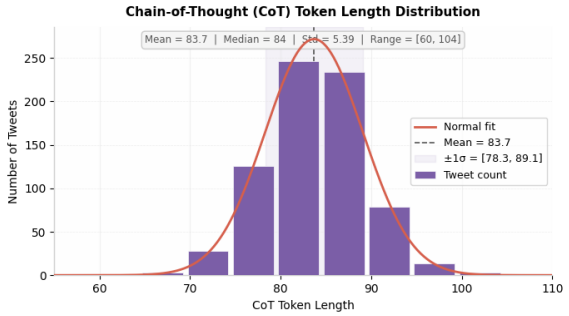


Figure 5: Distribution of budget-limited CoT token lengths before post-generation pruning. Mean = 83.7, $\sigma = 5.39$, with lengths ranging from 60–104 tokens.

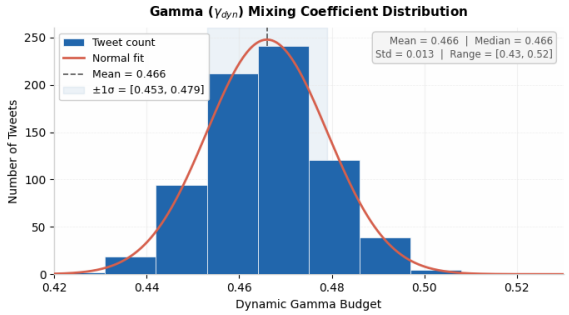


Figure 6: Distribution of γ_{dyn}, which exhibits limited variation across instances ($\sigma = 0.013$).

Figure 7 shows the distribution of the adaptive fusion weight α. The lower bound $[0.30, 0.40]$ is reached in 55.4% of inferences, while $\alpha > 0.5$ occurs in 41.3% of cases.

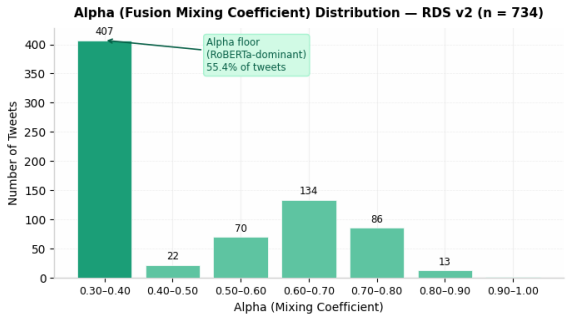


Figure 7: Distribution of the adaptive fusion weight α. The $[0.30, 0.40]$ range contains 55.4% of inferences, while $\alpha > 0.5$ occurs in 41.3%.

C.2 Symbolic Prior Activation

Figure 8 compares the symbolic prior activation across TweetEval and iSarcasm. The difference in activation rates illustrates the domain dependence of these explicit syntactic cues.

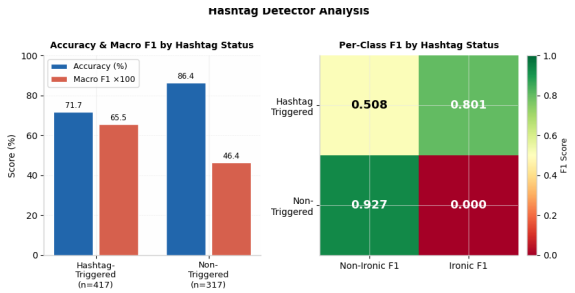


Figure 8: Hashtag activation analysis across TweetEval and iSarcasm, illustrating the domain dependence of explicit syntactic cues.

C.3 Cross-Domain Analysis

Figure 9 shows the reduction in false positives on iSarcasm, from 315 to 244.

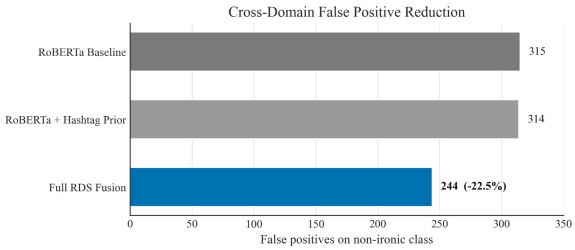


Figure 9: Cross-domain false-positive reduction on iSarcasm. RDS Fusion reduces false positives from 315 to 244.

Figure 10 shows the corresponding precision–recall trade-off for the ironic class, highlighting the precision-recovery behavior of RDS.

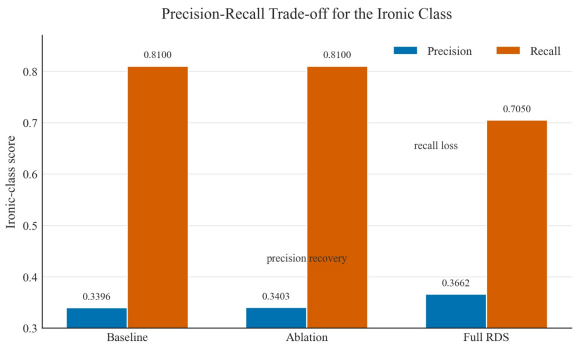


Figure 10: Precision–recall trade-off on the iSarcasm ironic class, highlighting precision recovery under cross-domain evaluation.

D Two More Case Studies

D.1 Case Study 2: Correct Irony Classification with Composite Signals (Tweet 172, TweetEval Dataset)

Tweet: *Yay for Fire Alarms at 3AM #not*

This tweet contains a strong explicit irony marker (`#not`), which the symbolic prior immediately identifies, assigning $S_{\text{prior}}(x) = \rho_{\text{strong}} = 0.88$. This caps the baseline trust weight at $\alpha \leftarrow \min(\alpha, 0.30)$.

RoBERTa baseline: The fine-tuned encoder independently assigns a very high irony probability ($P_{\text{RoBERTa}}(\hat{y} = 1|x) = 0.980$), consistent with the overtly positive framing ("Yay") applied to an objectively unpleasant event (a 3AM fire alarm). Both signals are strongly aligned.

CoT pipeline: The LLM reasoning trajectory outputs $P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) = 0.85$, independently parsing the pragmatic incongruence between the enthusiastic exclamation and the disruptive context. With entropy $H(c) = 0.342$ mapping to $\gamma_{\text{dyn}} = 0.469$, the Local Guardian whitelists critical tokens including *indicates*, *disruptive*, *excitement*, and *Tone*. The strong signal override recomputes $P_{\text{base}}(\hat{y} = 1|x) = 0.30 \times 0.980 + 0.70 \times 0.85 = 0.889$, and the final blend yields $P(\hat{y} = 1|x) = 0.60 \times 0.889 + 0.40 \times 0.88 = 0.885$. Correct Ironic prediction.

This instance demonstrates the framework's most coherent operating mode: the symbolic prior, the RoBERTa encoder, and the CoT pipeline are unanimously aligned, producing a high-confidence, correctly calibrated ironic prediction.

D.2 Case Study 3: Failure under Conflicting Signals (Tweet 143, TweetEval Dataset)

Tweet: *how is it possible that somebody so interesting is also so, so, so boring? #irony #what*

This is a ground-truth ironic tweet containing a definitive author-labeled marker (`#irony`). The symbolic prior correctly activates on this strong signal, assigning $S_{\text{prior}}(x) = 0.88$ and engaging the strong-signal override to cap the baseline trust weight at $\alpha = 0.300$.

RoBERTa baseline: The fine-tuned encoder assigns a severely low irony probability ($P_{\text{RoBERTa}}(\hat{y} = 1|x) = 0.087$), entirely failing to resolve the self-contradictory predication ("interesting" and "boring" applied to the same subject).

CoT pipeline: The compressed CoT output is $P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) = 0.15$ with a predictive entropy of $H(c) = 0.249$. The reasoning trajectory misinterprets the oxymoronic construction as a literal complaint, lacking the contextual agility to recognize the pragmatic inversion.

Adaptive Fusion: Because both neural components fail with extreme confidence, their combined base probability collapses to $P_{\text{base}}(\hat{y} = 1|x) = 0.30 \times 0.087 + 0.70 \times 0.15 = 0.131$. Even after applying the heavy strong-signal blend ($\omega_{\text{strong}} = 0.60$), the prior cannot rescue the prediction: $P(\hat{y} = 1|x) = 0.60 \times 0.131 + 0.40 \times 0.88 = 0.431$. The result falls below the 0.5 decision threshold, yielding a false negative.

This case exposes a critical architectural boundary: the limits of the symbolic prior's rescue capacity. When both the fine-tuned encoder and the compressed CoT confidently fail in unison, their combined bias mathematically overpowers even a perfectly fired, high-confidence explicit marker.

E Comparison of Unrestricted, Restricted and Post-pruned CoT Trajectories

To provide a qualitative view of the two-stage reasoning constraint, we examine a single held-out example (Tweet #51) under three inference configurations: unrestricted CoT, CoT restricted by the 120-token budget, and post-generation Local Guardian pruning of the restricted trajectory. All three configurations use the same manually authored prompt. The unrestricted configuration permits up to 1000 generated tokens and directly extracts the final score produced by the LLM, whereas the restricted and post-pruned configurations use the confidence-reconstruction procedure described in subsection 3.3.1.

Tweet: *Rangers league game with Alloa moved because of the Petrofac Cup final in April... #irony.*
True Label: $y = 1$ (ironic).

E.1 Unrestricted CoT

To illustrate the effect of the two-stage compression process, we first examine the model's unrestricted reasoning trajectory. With unrestricted generation, the model produced the following 321-word reasoning trajectory and assigned a raw LLM score of **0.05**:

Configuration	Length (words)	Score
Unrestricted CoT	321	0.05
Restricted CoT	86	0.69
Post-pruned CoT	47	0.57

Table 6: Three-way comparison of CoT trajectories for Tweet #51. The unrestricted configuration reports the raw LLM score, while the restricted and post-pruned configurations report reconstructed CoT signals.

CoT Generated by the LLM with `max_new_tokens = 1000`

Highlighting: Semantic evidence Decisive reasoning

Let's analyze the given tweet step by step:
Tweet Analysis:
Tweet: 'Rangers league game with Alloa moved because of the Petrofac Cup final in April... #irony'
Step-by-Step Reasoning:
1. **Literal Meaning vs. Real-World Situation:** - The literal meaning of the tweet is that a Rangers league game against Alloa was rescheduled due to the Petrofac Cup final. - There is no apparent contradiction between the literal meaning and the real-world situation. The tweet is stating a factual event without any indication of irony.
2. **Exaggerated Tone:** - The tweet does not use exaggerated language or phrases that would suggest irony. - Phrases like "moved because of the Petrofac Cup final in April" are straightforward and do not appear overly enthusiastic or negative.
3. **Hashtags:** - The hashtag "#irony" is present, which might suggest that the author intends for the tweet to be interpreted as ironic. - However, the presence of a hashtag alone is not enough to confirm irony; it often indicates a deliberate attempt to convey irony but doesn't always work.
4. **Mockery/Criticism:** - There is no clear mockery or criticism in the tweet. - The author is simply reporting a fact about a rescheduled game.
5. **Reader Interpretation:** - While a reasonable reader might interpret this as ironic, the lack of additional context or explicit indicators makes it difficult to confidently assert that the tweet is ironic.
6. **Word Usage:** - The tweet uses common phrases ("moved because of") without any elongated or exaggerated words that typically accompany irony.
Conclusion:
The tweet is straightforward and does not contain any clear indicators of irony. It is simply reporting a factual event without any apparent contradiction or exaggerated tone.
Score: Score: 0.05
This score reflects the very low likelihood that the tweet is ironic, given the absence of clear indicators and the straightforward nature of the information provided. The tweet is most likely a sincere statement about a rescheduled game.

The unrestricted trajectory explicitly recognizes the #irony marker but continues with substantially longer deliberation about whether the marker should be trusted, ultimately producing a low irony score. This example illustrates that a larger reasoning budget can lead to a materially different confidence assignment.

E.2 Restricted CoT

Applying the 120-token generation ceiling used by RDS Fusion produces the following 86-word trajectory:

CoT Restricted by `max_new_tokens = 120`. Tokens marked Green are Retained after post-pruning; tokens marked Red are Deleted.

To analyze the given tweet, let's break it down using the provided key signals:
Key Signals Analysis:
1. **Literal Meaning** vs. **Intended Meaning:**
- The tweet states: "Rangers league game with Alloa moved because of the Petrofac Cup final in April... #irony"
- The literal meaning is that a Rangers game against Alloa was rescheduled due to the Petrofac Cup final.
- The intended meaning, as indicated by the hashtag #irony, suggests that the reason for the rescheduling is ironic or not what it seems.

The restricted trajectory terminates before the extended reconsideration present in the unrestricted generation and yields a reconstructed CoT signal of **0.69**. The entropy of this sequence was found to be 0.24 and therefore γ_{dyn} was 0.448.

E.3 Post-Pruned CoT

Local Guardian is then applied to the restricted trajectory using the target-specific retention rule described in subsection 3.3.3. Therefore, a pruned chain-of-thought compressed by the ratio $\gamma_{\text{dyn}}(=0.448$ in this case) is obtained. The resulting 47-word trajectory is:

Pruned CoT, Used by RDS Fusion: Retained and Critical Tokens

analyze given tweet, let's break using provided key signals: **Key Signals** Analysis: **Literal Meaning Intended Meaning:** tweet states: "Rangers league game Alloa moved Petrofac Cup final April... #irony" literal meaning Rangers game Alloa rescheduled due Petrofac Cup final. intended, meaning, indicated hashtag #irony, suggests ironic not

The post-pruned trajectory yields a reconstructed CoT signal of **0.57**. The critical tokens identified for this example are `ironic`, `Analysis`, `indicated`, `suggests`, and `intended`. Notably, the retained trajectory continues to contain the literal-versus-intended-meaning distinction and the explicit ironic cue despite substantial token removal.

E.4 Analysis

Overall, the example illustrates how the successive transformations introduced by the inference pipeline can alter the semantic evidence available for classification. The ground-truth label of the tweet is ironic ($y = 1$), whereas the unrestricted 321-word reasoning trajectory incorrectly interprets the tweet as non-ironic and assigns a raw confidence of only 0.05. Applying the generation-time budget substantially shortens the trajectory to 86 words; although the explicit confidence information may be removed during this restriction, the surviving reasoning provides sufficient semantic evidence for confidence reconstruction, yielding a score of 0.69, which is on the correct ironic side of the decision boundary. The subsequent Local Guardian performs a more targeted token-level reduction, retaining the most task-relevant tokens and reducing the trajectory further to 47 words. The reconstructed confidence remains above the decision threshold at 0.57, thereby preserving the correct prediction despite the substantially shorter reasoning trajectory.

The example therefore illustrates two important effects of the proposed compression strategy. Compression does not merely reduce the length of the reasoning trajectory; it can also change the semantic evidence available for downstream confidence extraction. On this particular TweetEval test-set, unrestricted CoT yields the worst performance, marginally below the restricted trajectory and 2.6% below the restricted and post-pruned CoT. The progression from 0.05 to 0.69 and subsequently to 0.57 should not, however, be interpreted as evidence that shorter reasoning is always superior. Rather, it provides a qualitative example of how successive compression stages can remove distracting or misleading reasoning while retaining sufficient task-relevant evidence to recover and preserve the correct prediction.

F Hardware, Environment, and Reproducibility Details

F.1 Hardware Configuration

All experiments were executed on Kaggle and Google Colab environments with NVIDIA dual T4 GPU acceleration (2×16 GB VRAM, CUDA 12.2). The TweetEval evaluation ($N = 734$) required approximately 2.8 hours. The 120-token generation ceiling prevents arbitrarily long autoregressive reasoning; post-generation pruning produces the compact trajectory used downstream.

F.2 Generation Parsing Guardrails

The following secondary guardrails were applied to CoT outputs and predictive calculations before final confidence extraction:

1. **Encoding and Formatting Standardization:** Model generations are decoded with `skip_special_tokens=True` and standardized via lowercase conversion and whitespace stripping prior to parsing.

2. **Score Boundary Clamping:** Any extracted continuous confidence score is mathematically bounded using a min-max clamp to ensure it falls within the valid $[0.0, 1.0]$ probability range.

3. **Unresolvable Ambiguity Fallback:** If the hierarchical regex and semantic parsers fail to identify a definitive score within the generated text, the extraction naturally defaults to $P_{\text{CoT}}(\hat{y} = 1|\tilde{c}_{\text{RDS}}) = 0.5$. This explicitly triggers the two-pass fallback mechanism, which re-prompts the model for a forced verdict.

4. **Entropy Underflow Prevention:** During the predictive entropy $H(c)$ computation, an epsilon value ($= 10^{-9}$) is added to the token probabilities prior to the logarithmic transformation to prevent undefined `log(0)` errors. A secondary `NaN` catch enforces a 0.0 entropy baseline if precision limits are exceeded.

F.3 Quantization Details

`Qwen2.5-3B-Instruct` was loaded using 4-bit NF4 (Normal Float 4) quantization via `bitsandbytes` to optimize memory footprint:

- **Quantization type:** `nf4` (NormalFloat 4-bit, optimized for normally distributed weights (Yang et al., 2024)).

- **Double quantization:** enabled (quantizes the quantization constants to further reduce memory).
- **Compute dtype:** `float16`.
- **Full model (RDS Fusion) memory footprint (post-quantization):** $\approx$ 2.1 GB VRAM.
- **Generation (RDS Fusion):** greedy decoding (`temperature=0`, `do_sample=False`), `max_new_tokens=120`.
- **Generation (Unrestricted Reasoning):** greedy decoding (`temperature=0`, `do_sample=False`), `max_new_tokens=1000`.
- **Two-pass fallback:** same model, `max_new_tokens=10`.

F.4 Validation Protocol and Anti-Leakage Guarantee

To prevent data leakage, the first 50 tweets of the TweetEval test set were designated as a validation split *before* any parameter search. All calibration of H_{min}, H_{max}, α_{min}, α_{max}, ρ_{strong}, and blending weights was strictly confined to these 50 samples (achieving an over-fitted validation accuracy of 88.0% on this calibration split). The remaining 734 tweets were treated as a completely unseen held-out test set. The iSarcasm evaluation used frozen TweetEval calibrations with zero additional tuning, guaranteeing a strictly few-shot cross-domain evaluation.

F.5 Consolidated Calibration Thresholds

Table 7 consolidates all static thresholds and fusion parameters derived from the 50-tweet validation split.

Module	Parameter	Value
Compression Budget	$H_{\text{min}}, H_{\text{max}}$	0.2, 0.5
Compression Bounds	$\gamma_{\text{min}}, \gamma_{\text{max}}$	0.1, 0.9
Retention Overrides	$\tau_{\text{crit}}, \tau_{\text{conn}}$	1000, 500
Retention Overrides	$\tau_{\text{cont}}, \tau_{\text{low}}$	10, 1
Prior Weights	$\rho_{\text{strong}}, \rho_{\text{cap}}$	0.88, 0.70
Weak Prior Weights	$\rho_{\text{weak}}, \rho_{\text{emoji}}, \rho_{\text{elong}}$	0.15, 0.25, 0.12
Fusion Bounds	$\alpha_{\text{min}}, \alpha_{\text{max}}$	0.45, 0.90
Skepticism Gate	$H_{\text{sk}}, \lambda_{\text{sk}}, p_{\text{sk}}$	0.25, 0.5, 0.15
Blend Weights	$\omega_{\text{strong}}, \omega_{\text{weak}}$	0.60, 0.75

Table 7: Consolidated pipeline hyperparameters.

F.6 Library Versions

Library	Version
`transformers`	4.40.0
`torch`	2.2.0+cu121
`bitsandbytes`	0.43.1
`llmlingua`	0.2.2
`numpy`	1.26.4
`scikit-learn`	1.4.2
`statsmodels`	0.14.1

Table 8: Library versions for reproducibility.

RSPC: A Benchmark for Modeling Stress and Psychiatric Conditions in Digitally Mediated Relationships using Psychiatrist Annotations

Parmitha Vangapandu[1,*], **Sai Ganesh Mokkapati**[2], **Sathwik Narkedimilli**[3], **MSVPJ Sathvik**[4], **Timothy Liu**[5], **Simon See**[5], and **Johannes C. Eichstaedt**[6,7,*]
[1]Indian Institute of Information Technology Dharwad
[2]Keshav Memorial Engineering College
[3]National University of Singapore, [4]University of Birmingham
[5]NVIDIA, Singapore, [6]Stanford University, [7]INSEAD
[*]Corresponding authors
23bcs092@iiitdwd.ac.in, mokkapatisaiganesh@gmail.com,
sathwik.narkedimilli@ieee.org, msvpjsathvik@gmail.com,
timothyl@nvidia.com, ssee@nvidia.com
Correspondence: johannes.stanford@gmail.com | **Project Website:** Link

Abstract

In NLP, mental health conditions are often modeled as isolated phenomena, without interpersonal context. We use Reddit posts about long-distance relationships to capture both mental health distress and associated relational triggers. We introduce the Relational Stress and Psychiatry Corpus (RSPC), containing 1,799 Reddit posts annotated by psychiatrists for diagnostic categories, including the most prevalent mood disorders (anxiety and depression), relational stressor triggers, and indications of relationship phase. We benchmark seven fine-tuned transformer models and five large language models across the tasks of multi-label disorder classification, relational trigger detection, and temporal phase prediction. We find clear task-dependent differences between model families, with Claude-3-Haiku achieving the best disorder classification performance (Macro-F1 = 0.538) and GPT-4o obtaining the strongest relational trigger detection performance (Macro-F1 = 0.519), suggesting distinct model capabilities. We further find strong associations between anxiety disorders and chronic relational uncertainty. Overall, RSPC establishes a benchmark for NLP tasks that consider relational context and supports a shift from individual-centric to context-aware mental health modeling, capturing the social and temporal dynamics of distress.

1 Introduction

As global mobility increases, digital platforms mediate intimate relationships, profoundly impacting attachment and separation. However, computational mental health predominantly models psychiatric distress as an individual-level phenomenon (treating symptoms as isolated internal states independent of external social triggers). This ignores the fact that relational context is vital: symptoms emerge from recurring stressors such as communication silences, jealousy, commitment ambiguity, and reunion-separation cycles. We must therefore shift toward relationally grounded natural language processing frameworks. Clinical studies firmly associate relationship-mediated stress with severe anxiety, adjustment difficulties, clinical depression, and insomnia (Stafford, 2004; Neustaedter and Greenberg, 2012).

Prior datasets largely fail to capture this complexity, reducing distress to coarse binary classification (Coppersmith et al., 2015; Losada et al., 2018; Yates et al., 2017; Cohan et al., 2018). They ignore the diverse mood and anxiety subtypes defined in DSM-5-TR, relying heavily on user-level annotations that strip away surrounding relational context. Furthermore, these benchmarks consistently operationalize temporality merely as chronological time (modeling time strictly via linear posting histories or absolute timestamps). This mechanistic approach entirely misses the underlying developmental phases of a relationship, preventing models from detecting how specific temporal transitions trigger escalating psychiatric symptoms over extended observational periods.

To address this, we focus on long-distance relationships. LDRs exclusively rely on digital communication, eliminating unrecorded physical interactions and providing an ideal context for observing relationally mediated distress. Unlike prior datasets that used coarse MTurk crowdsourcing, we collaborated with four psychiatrists from

Proceedings of the 11th Workshop on Natural User-generated Text, pages 286–312
October 28, 2026 ©2026 Association for Computational Linguistics

Andhra University to develop a schema aligned with the DSM-5-TR (Association, 2022) and ICD-11 (World Health Organization et al., 2018) criteria. Four trained annotators independently evaluated each post for symptoms, stressors, and temporal phases. Disagreements were adjudicated strictly, yielding substantial inter-annotator agreement and high reliability across all evaluated clinically informed annotation tiers.

Leveraging this framework, we introduce the **Relational Stress and Psychiatry Corpus (RSPC)**, comprising 1,799 Reddit posts from long-distance communities annotated for DSM-5-TR/ICD-11-aligned symptoms, relational stressors, and temporal phases. Our contributions are threefold: (1) we introduce the first benchmark linking clinically grounded psychiatric categories with relational stressors and temporal phases in digitally mediated relationships, (2) we benchmark transformers and LLMs across disorder classification, trigger detection, and temporal reasoning tasks, and (3) we demonstrate that relational context provides a measurable signal for psychiatric inference within communication environments.

2 Related Work

Recent work on **Mental Health Detection in Social Media** has primarily focused on detecting depression, suicidality, and anxiety from social media language (Coppersmith et al., 2015; Yates et al., 2017; Cohan et al., 2018; Losada et al., 2018). Earlier approaches relied on lexical features, whereas recent work has adopted transformer architectures (Nadeem, 2016; Harrigian et al., 2021). Large language models have further expanded the scope of zero-shot psychiatric inference. Existing benchmarks, however, largely model psychiatric distress as an individual-level attribute detached from interpersonal context. Relational dynamics such as communication disruption, attachment insecurity, and jealousy remain underrepresented despite their clinical relevance.

Relational and Contextual Modeling. Prior work has explored conversational dynamics (Alghowinem et al., 2016), online social support (De Choudhury et al., 2013), emotional disclosure (Pavlova and Berkers, 2020), and longitudinal behavioral change (MacAvaney et al., 2018; Sekulić et al., 2018). While recent studies extract life events and social contexts to explain depression (Chen et al., 2024; Lv et al., 2025), these approaches still largely emphasize broader environmental factors over clinically meaningful interpersonal stressors, such as communication gaps or commitment ambiguity. No prior work explicitly links event-contingent relational stressors with DSM-5-TR psychiatric symptom categories.

Clinical Grounding and Ethics. Recent studies highlight limitations in clinical grounding, including noisy self-disclosure labels, inconsistent annotations, and weak alignment with diagnostic frameworks (Chancellor et al., 2019; Harrigian et al., 2021). Researchers increasingly advocate for clinically informed annotation protocols and collaboration with mental health professionals (Low et al., 2020; Zirikly et al., 2019). Prior datasets, however, predominantly rely on coarse binary depression labels rather than differentiated symptom-level psychiatric categories in relational settings.

Large Language Models for Mental Health. Recent work explores LLM prompting for psychiatric classification, counseling simulation, and risk detection (Guo et al., 2024; Gilardi et al., 2023). Prompted LLMs often perform strongly on socially contextual tasks, but their behavior under clinically grounded multi-label psychiatric inference remains underexplored. Existing work also lacks a systematic comparison between fine-tuned transformers and prompted LLMs for relationally grounded psychiatric reasoning.

Research Gap & Positioning of RSPC: Existing mental health NLP research predominantly models psychiatric distress as an individual-level phenomenon using coarse diagnostic labels, with limited consideration of the relational and interpersonal dynamics through which symptoms emerge. Prior work on social-context modeling has focused primarily on generic interaction patterns or longitudinal behavior, while clinically meaningful relational stressors, such as communication disruption, attachment insecurity, and commitment ambiguity, remain underexplored. Furthermore, existing benchmarks lack DSM-5-TR-aligned multi-label psychiatric annotations and systematic evaluation of LLM-based relational psychiatric reasoning. RSPC addresses these gaps through clinically grounded relational-context modeling and comprehensive benchmarking of transformers and LLMs for psychiatric symptom inference.

3 RSPC CONSTRUCTION

This section presents the methodology and data-collection pipeline used to construct RSPC as described in Fig. 1.

3.1 Dataset Collection & Construction

We introduce the **Relational Stress and Psychiatry Corpus (RSPC)**, a benchmark for studying the expression of psychiatric symptoms in digitally mediated romantic relationships. The dataset comprises publicly available Reddit posts from long-distance relationship communities (`r/LongDistance`, `r/LDR`) collected between January 2020 and December 2023 and filtered for narrative completeness, relational relevance, and linguistic consistency. All usernames, personal identifiers, and proper nouns were anonymized using placeholder tokens (e.g., `[USER]`, `[PLACE]`). Consistent with prior mental health NLP research, only publicly accessible posts were used, with no attempts made to infer user identities or contact individuals directly.

3.2 Data Annotation & Guidelines

The annotation framework was developed in consultation with a team of four licensed psychiatrists from Andhra University and grounded in DSM-5-TR (Association, 2022) and ICD-11 (World Health Organization et al., 2018) diagnostic criteria. Each Reddit post was independently annotated by four trained annotators using a clinically informed coding manual, with disagreements resolved through adjudication. Inter-rater reliability, measured using Cohen's κ, demonstrated substantial agreement across all annotation tiers, including psychiatric symptoms (0.78), relational stressors (0.72), and temporal relationship phases (0.81).

1. **Tier A: Psychiatric Symptom Categories.** Posts were annotated for five clinically grounded symptom categories: Major Depressive Disorder (MDD), Generalized Anxiety Disorder (GAD), Separation Anxiety Disorder (SAD), Adjustment Disorder (ADJ), and Insomnia, using DSM-5-TR/ICD-11-aligned criteria adapted for textual inference.

2. **Tier B: Relational Stressor Triggers.** Posts were annotated for relational stressors, including Commitment Ambiguity, Lack of Communication, Reunion/Separation Stress, Trust/-Fidelity Issues, Jealousy/Insecurity, Silence Gaps, Social Media Surveillance, and Timezone Misalignment.

3. **Tier C: Temporal Relationship Phase.** Posts were assigned to one of four temporal phases: SEPARATION, ANTICIPATION, REUNION, or UNKNOWN, modeling event-contingent rather than chronological time.

Table 1: Psychiatric symptom label prevalence across splits.

Split	n	ADJ	GAD	SAD	MDD
Train	1,259	74.7%	71.3%	42.7%	17.1%
Validation	179	77.1%	69.8%	48.6%	16.8%
Test	361	72.9%	70.9%	43.8%	16.9%
Full	**1,799**	**74.5%**	**71.1%**	**43.5%**	**17.0%**

Note: Insomnia prevalence is 0.9%, 1.1%, 2.2%, and 1.2% for Train, Validation, Test, and Full, respectively.

Table 2: Relational trigger prevalence across dataset splits (%).

Split	CA	LC	RS	TF	J
Train	67.0	60.4	21.5	17.7	18.0
Validation	67.0	66.5	21.8	22.9	15.6
Test	62.9	64.0	21.9	18.0	19.4
Full	**66.2**	**61.7**	**21.6**	**18.3**	**18.0**

Note: Full-corpus prevalence for the less frequent triggers SG, SMS, and TZM is 5.7%, 2.1%, and 1.6%, respectively.

3.3 Dataset Statistics

The final corpus contains 1,799 Reddit posts, split into train/validation/test partitions with a 70:10:20 stratified ratio. Adjustment Disorder (74.5%) and GAD (71.1%) are the dominant psychiatric categories, while MDD (17.1%) and Insomnia (1.2%) remain sparse. Commitment Ambiguity (66.2%) and Lack of Communication (61.7%) are the most common relational stressors. Temporal phase labels are dominated by SEPARATION (65.1%). Table 1 and Table 2 report full per-split label distributions; co-occurrence statistics are provided in Appendix A.

3.4 Inter-annotator Agreement Scores

Inter-annotator agreement was evaluated using Cohen's κ across all annotation tiers, demonstrating substantial agreement for psychiatric symptoms, relational triggers, and temporal phase annotations. A detailed analysis of the agreement and tier-wise scores is provided in Appendix 7.

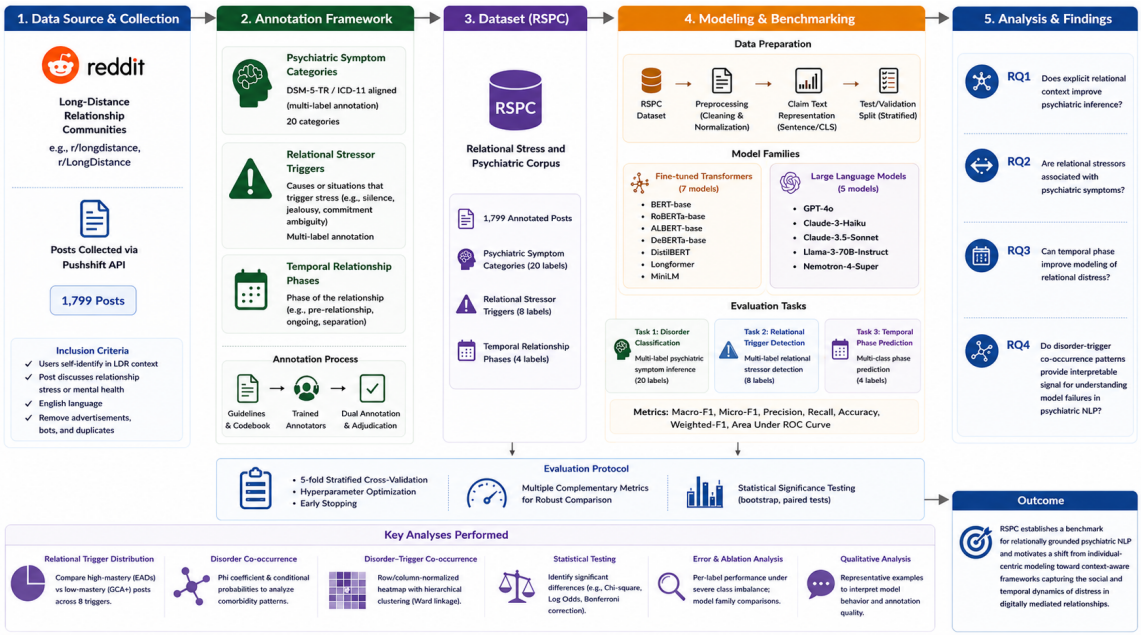


Figure 1: Workflow diagram of RSPC

4 Methodology

This section presents the major experiments conducted on **RSPC**, involving LLM- and PLM-based approaches. Detailed experimental settings and results are described in the following subsections.

4.1 Models and Experimental Setup

We evaluate RSPC across three benchmark tasks: multi-label psychiatric symptom classification, relational trigger detection, and temporal phase classification. Experimental comparisons examine supervised representation learning against instruction-following inference under clinically grounded relational settings. Seven transformer architectures are benchmarked, including BERT-base (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019), ClinicalBERT (Alsentzer et al., 2019), BART-base (Lewis et al., 2020), T5-base (Raffel et al., 2020), Longformer (Beltagy et al., 2020), and BigBird-RoBERTa (Zaheer et al., 2020). Additionally, five prompted large language models are evaluated: GPT-4o (Achiam et al., 2023), Claude-3-Haiku (Anthropic, 2024), Qwen-2.5-72B (Hui et al., 2024), LLaMA-3-70B (Touvron et al., 2023), and Nemotron-Super (Adler et al., 2024). These experiments analyze contextual reasoning, long-document understanding, and clinically informed relational inference across heterogeneous architectures within RSPC benchmarks.

4.2 Hyperparameters

Transformer architectures were fine-tuned independently for each benchmark task using the AdamW optimizer (Loshchilov and Hutter, 2017). Training used a learning rate of 2×10^{-5}, a batch size of 16, and early stopping based on validation Macro-F1 performance. Multi-label classification tasks utilized sigmoid activation with binary cross-entropy loss and inverse-frequency class weighting for psychiatric symptom and relational trigger categories. Fine-tuning procedures were standardized across all transformer encoders to ensure fair architectural comparisons. Additional implementation details, optimization settings, and task-specific hyperparameters are provided in Appendix A for complete experimental reproducibility and transparency.

4.3 Prompts & Prompting Strategies

Large language models were evaluated under deterministic prompting configurations using temperature $T = 0.0$ to minimize stochastic output variability across experimental runs. Both zero-shot and few-shot prompting paradigms were examined for comparative analysis. Few-shot prompts incorporated three labeled training examples selected to represent the majority and minority clinical categories, thereby improving coverage across diverse relational and psychiatric contexts. Prompt engineering procedures remained consistent across evaluations of GPT-4o, Claude-3-Haiku, Qwen-2.5-72B, LLaMA-3-70B, and Nemotron-Super. Com-

plete prompting templates, instruction formats, and demonstration examples are provided in Appendix A to support methodological transparency, reproducibility, and consistent evaluation settings and protocols.

4.4 Evaluation

Performance evaluation employed multiple complementary metrics implemented using scikit-learn (Pedregosa et al., 2011). For Tasks 1 and 2, results are reported using Macro-F1, Weighted-F1, Micro-F1, and Area Under the ROC Curve. Macro-F1 served as the primary evaluation metric because it emphasizes minority psychiatric categories and mitigates the dominance of the majority class. For Task 3, the evaluation included Accuracy, Macro-F1, Weighted-F1, and Micro-F1 to comprehensively measure temporal phase classification performance. Metric selection was designed to capture balanced predictive performance, clinically relevant sensitivity for minority groups, and overall classification robustness across heterogeneous relational mental-health prediction tasks.

5 Experiments & Discussion

5.1 Task & Research Question Formulation

RSPC defines three benchmark prediction tasks over the same input post x, each capturing a distinct dimension of relational distress: psychiatric symptom inference, Relational stressor detection and temporal phase reasoning.

1. **Task 1 – Multi-Label Disorder Classification:** Models predict one or more psychiatric symptom categories expressed or implied within a post. **RQ1:** *Can transformers and LLMs infer clinically grounded psychiatric categories from LDR narratives?*

2. **Task 2 – Relational Trigger Detection:** Models identify relational stressors associated with psychological distress. **RQ2:** *Do relational stressors provide diagnostic value beyond symptom-only text?*

3. **Task 3 – Temporal Phase Classification:** Models predict the temporal relationship phase described in the post. **RQ3:** *Can temporal phase improve modeling of relational distress?*

4. **Cross-Task Analysis. RQ4:** *Do disorder-trigger co-occurrence patterns provide interpretable signal for understanding model failures in psychiatric NLP?* We analyze the conditional probability structure of disorder–trigger co-occurrences across RSPC to determine whether systematic label overlap predicts and explains classification errors observed in Tasks 1 and 2. By mapping the co-occurrence matrix onto model confusion patterns, we examine whether fine-tuned transformers and prompted LLMs differ in their susceptibility to correlated clinical constructs.

5.2 Task-1 - RQ1: Inferring Psychiatric Disorders

Task 1 evaluates multi-label psychiatric disorder classification to address **RQ1**, which examines whether transformers and LLMs can infer clinically grounded psychiatric categories from long-distance relationship narratives. Results in Table 3 show that BigBird-RoBERTa achieves the strongest Macro-F1 among fine-tuned transformers (0.505), while Claude-3-Haiku attains the best overall Macro-F1 (0.538) under zero-shot evaluation. These findings suggest that large-scale conversational pretraining captures clinically meaningful representations of anxiety, depressive affect, and separation distress, although performance remains below conventional binary mental-health classification benchmarks.

Per-label results are provided in Appendix A. Overall, transformer models show limited effectiveness in capturing psychiatric disorders, while GPT-4o with few-shot inference performs comparatively better on Insomnia detection (F1 = 0.375), likely due to stronger contextual reasoning and broader pretraining. Higher performance across most categories, such as ADJ, GAD, and SAD, further indicates that supervised transformers benefit from well-represented clinical linguistic patterns.

5.3 Task-2 - RQ2: Relational Stressors

Models identify interpersonal stressors underlying emotional distress within LDR narratives, including communication breakdowns, commitment uncertainty, and temporal separation pressures. **RQ2:** *Do relational stressors provide diagnostic value beyond symptom-only text?* Table 4 shows that GPT-4o few-shot achieves the highest Macro-F1 (0.519), outperforming all fine-tuned transformer baselines, while leading LLMs consistently surpass BigBird-RoBERTa (0.478), the strongest supervised encoder. Unlike Task 1, these findings indicate that relational trigger detection depends

Table 3: Task 1: Multi-Label Disorder Classification. The best overall results are shown in **bold**; best transformer results are underlined.

Model	Strategy	Macro-F1	Wt.-F1	Micro-F1	AUC
Longformer	fine-tuned	0.450	0.601	0.591	0.607
BART-base	fine-tuned	0.496	0.671	0.658	0.625
T5-base	fine-tuned	0.461	0.649	0.646	0.520
ClinicalBERT	fine-tuned	0.470	0.654	0.628	0.545
RoBERTa-base	fine-tuned	0.486	0.661	0.656	0.668
BERT-base	fine-tuned	0.504	**0.699**	**0.687**	0.634
BigBird-RoBERTa	fine-tuned	0.505	0.684	0.682	0.625
GPT-4o	few-shot	0.438	0.521	0.498	0.613
GPT-4o	zero-shot	0.452	0.475	0.521	0.598
Nemotron-Super	few-shot	0.239	0.195	0.196	0.487
Nemotron-Super	zero-shot	0.112	0.101	0.107	0.519
Qwen-2.5-72B	few-shot	0.482	0.553	0.582	0.620
Qwen-2.5-72B	zero-shot	0.423	0.550	0.573	0.589
LLaMA-3-70B	few-shot	0.503	0.582	0.596	0.622
LLaMA-3-70B	zero-shot	0.451	0.603	0.607	0.601
Claude-3-Haiku	zero-shot	**0.538**	0.681	0.668	0.610
Claude-3-Haiku	few-shot	0.483	0.566	0.567	0.592

less on stable lexical symptom patterns and more on socially grounded reasoning, pragmatic interpretation, and conversational knowledge transferred through large-scale pretraining.

Table 4: Task 2: Relational Trigger Detection. The best overall results are shown in **bold**; best transformer results are underlined.

Model	Strategy	Macro-F1	Wt.-F1	Micro-F1	AUC
T5-base	fine-tuned	0.189	0.317	0.404	0.570
BART-base	fine-tuned	0.346	0.583	0.584	0.709
ClinicalBERT	fine-tuned	0.388	0.588	0.572	0.703
BERT-base	fine-tuned	0.392	0.582	0.568	0.725
Longformer	fine-tuned	0.424	0.587	0.572	0.723
RoBERTa-base	fine-tuned	0.450	0.611	0.589	0.728
BigBird-RoBERTa	fine-tuned	0.478	0.605	0.591	0.734
Nemotron-Super	zero-shot	0.317	0.308	0.308	0.607
Nemotron-Super	few-shot	0.368	0.480	0.436	0.641
Claude-3-Haiku	zero-shot	0.419	0.602	0.534	0.716
Claude-3-Haiku	few-shot	0.470	0.630	**0.597**	0.729
Qwen-2.5-72B	zero-shot	0.483	0.612	0.571	**0.748**
Qwen-2.5-72B	few-shot	0.434	0.570	0.529	0.701
LLaMA-3-70B	zero-shot	0.465	0.605	0.586	0.728
LLaMA-3-70B	few-shot	0.492	0.587	0.567	0.736
GPT-4o	zero-shot	0.496	0.606	0.585	0.739
GPT-4o	few-shot	**0.519**	**0.648**	0.576	0.743

5.4 RQ2 Continued: Does Trigger Information Improve Disorder Classification?

Table 4 reports Task 2 (trigger detection) performance in isolation but does not directly test **RQ2** – whether relational Stressors improve disorder classification (Task 1). We address this directly by augmenting the Task 1 input with Task 2 trigger labels and measuring the resulting change in Macro-F1, using the LoRA-fine-tuned Qwen-2.5-7B classifier introduced in §5.1 as a consistent backbone across all three conditions.

Table 5: RQ2: Trigger-label augmentation on Task 1.

Setting	F1	AUC	ΔF1
Text only	.447	.598	–
Predicted triggers	.460	.602	+.013
Oracle triggers	.528	.736	+.081

Oracle-trigger augmentation, an upper bound because it uses ground-truth Task 2 labels unavailable at inference, improves Macro-F1 by +0.081, confirming that relational trigger information provides diagnostic value for RQ2. Predicted-trigger augmentation, using our Task 2 classifier's imperfect predictions (Macro-F1 = 0.313) rather than oracle labels, produces a smaller but still real gain of +0.013. Thus, the 0.068 Macro-F1 gap between oracle and predicted conditions reflects limitations in trigger classification rather than augmentation itself, highlighting Task 2 improvement as the most direct path toward practical benefit.

This pattern is supported by AUC: oracle augmentation substantially increases Macro-AUC from 0.598 to 0.736, whereas predicted-trigger augmentation leaves it nearly unchanged, from 0.598 to 0.602. This indicates that the modest, realistic F1 improvement does not correspond to a comparably stronger probabilistic ranking of disorder labels. Notably, Insomnia AUC declines from 0.496 to 0.434 under predicted-trigger augmentation, suggesting that the trigger classifier's sparse training signal for this label, with only 11 positive examples, may introduce noise rather than useful information for the rarest disorder category.

5.5 Task-3 - RQ3: Temporal Modelling

Models identify the relational phase represented within each LDR narrative, including SEPARATION, ANTICIPATION, REUNION, and UNKNOWN. **RQ3:** *Can temporal phase improve modeling of relational distress?* Table 6 reports temporal phase classification performance across transformer and prompted LLM architectures. Although fine-tuned transformers achieve high Accuracy, substantially lower Macro-F1 scores indicate strong majority-class bias due to the dominance of the SEPARATION phase (65.1%). Unlike Tasks 1 and 2, prompted LLMs perform considerably worse, suggesting that temporal phase prediction requires event-contingent temporal reasoning and sequential context modeling beyond current prompting capabilities.

Table 6: Task 3: Temporal Phase Classification. The best overall results are shown in **bold**; best transformer results are underlined.

Model	Strategy	Accuracy	Macro-F1	Wt.-F1	Micro-F1
Longformer	fine-tuned	0.920	0.445	0.901	0.920
BART-base	fine-tuned	0.934	0.495	0.921	0.934
T5-base	fine-tuned	0.909	0.432	0.889	0.909
ClinicalBERT	fine-tuned	0.922	0.458	0.907	0.922
RoBERTa-base	fine-tuned	0.939	**0.521**	**0.928**	0.939
BERT-base	fine-tuned	0.936	0.504	0.924	0.936
BigBird-RoBERTa	fine-tuned	**0.942**	0.516	0.929	**0.942**
GPT-4o	zero-shot	0.410	0.315	0.486	0.410
GPT-4o	few-shot	0.601	0.291	0.563	0.601
Claude-3-Haiku	zero-shot	0.575	0.268	0.538	0.575
Claude-3-Haiku	few-shot	0.485	0.294	0.537	0.485
Qwen-2.5-72B	zero-shot	0.615	0.351	0.591	0.615
Qwen-2.5-72B	few-shot	0.618	0.305	0.580	0.618
LLaMA-3-70B	zero-shot	0.643	0.320	0.604	0.643
LLaMA-3-70B	few-shot	0.665	0.374	0.613	0.665
Nemotron-Super	zero-shot	0.489	0.221	0.451	0.489
Nemotron-Super	few-shot	0.571	0.360	0.580	0.571

5.6 RQ4: Disorder-Trigger Co-occurrence as a Lens on Model Failures

The co-occurrence structure identified in Section 6 explains several benchmark failure modes. The complete SAD-ADJ overlap ($P = 1.00$) clarifies the confusion in Task 1, where models fail to separate linguistically similar labels and are biased toward the higher-prevalence ADJ class. In Task 2, the strong GAD-Commitment Ambiguity association (68.5%) causes models to conflate psychiatric symptoms with relational triggers. Similarly, the MDD-Trust/Fidelity coupling (14.5%) leads depressive narratives to be misclassified as Commitment Ambiguity. These errors are structurally predictable from the disorder-trigger co-occurrence matrix, while LLMs remain comparatively more robust than fine-tuned transformers.

6 Key Insights and Observations from RSPC Dataset

6.1 Relational Trigger Profiles Across Anxiety Groups

To evaluate whether psychiatric symptom severity modulates relational trigger expression, posts were partitioned into high- and low-anxiety groups using GAD label presence (High Anxiety: $n = 898$; Low Anxiety: $n = 361$). Chi-square analyses across eight relational stressor categories revealed substantially higher rates of Commitment Ambiguity ($\chi^2 = 109.70$, $p < .001$) and Lack of Communication ($\chi^2 = 108.93$, $p < .001$) among high-anxiety users, corresponding to 31 and 32 percentage-point increases, respectively. Reunion/Separation Stress was significantly elevated within the low-anxiety group ($\chi^2 = 31.98$, $p < .001$; $+14\%$). Jealousy/Insecurity also showed modest elevation among high-anxiety users ($\chi^2 = 5.78$, $p < .05$; $+6\%$). The remaining categories showed no significance.

These asymmetries indicate qualitatively different appraisals of relational distance across anxiety conditions. Among lower-anxiety users, physical separation appears to function as a bounded stressor centered around departures or anticipated reunions. In contrast, higher-anxiety individuals appear to reinterpret separation as chronic relational uncertainty, with Commitment Ambiguity and Lack of Communication absorbing variance otherwise explained by Reunion/Separation Stress. This pattern aligns with intolerance-of-uncertainty models of GAD (Dugas et al., 1998; Carleton, 2016), which posit that anxious individuals generalize situational stressors into diffuse threats to relational security, transforming logistical separation into perceived relational deterioration and existential vulnerability.

6.2 Disorder Co-occurrence Structure

We quantify psychiatric label co-occurrence using conditional probabilities $P(\text{col} \mid \text{row})$ instead of Phi coefficients. Although Phi equals Pearson's r for binary variables, it can be artificially attenuated or even become negative when prevalence is imbalanced (Warrens, 2008). Within RSPC, GAD occurs in 71.1% of posts, whereas MDD appears in 17.1%, producing misleading Phi estimates despite 188 genuine co-occurrences and substantial dependence ($P(\text{GAD} \mid \text{MDD}) = 0.62$). Conditional probabilities mitigate this distortion by normalizing co-occurrence frequencies relative to the prevalence of the row disorder across all observed samples.

Figure 3 demonstrates asymmetry across disorders. SAD exhibits near-complete overlap with ADJ ($P = 1.00$) and overlap with GAD ($P = 0.85$), indicating that separation anxiety in LDR settings rarely appears independently of anxiety-adjustment syndromes. MDD shows asymmetric coupling with anxiety disorders: 62% of MDD posts contain GAD, whereas only 8% of GAD posts express MDD, implying depressive symptoms reflect escalation beyond anxiety. Insomnia ($n = 21$) co-occurs strongly with ADJ (0.93), GAD (0.56), and MDD (0.20), consistent with rumination-based hyperarousal models and CBT-I (Harvey, 2002; Espie, 2007; Morin, 1993).

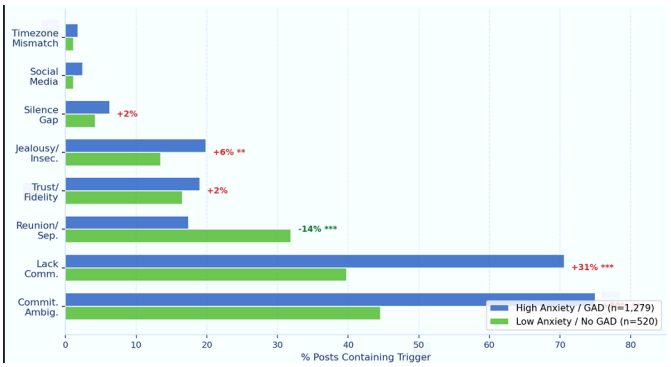


Figure 2: Relational trigger distribution across high-anxiety (GAD-positive) and low-anxiety (GAD-negative) groups. Bars represent the percentage of posts containing each trigger. Significant differences are annotated ($*$ $p < .05$; $***$ $p < .001$). Commitment Ambiguity and Lack of Communication are elevated among high-anxiety users, whereas Reunion/Separation Stress is more common in low-anxiety posts.

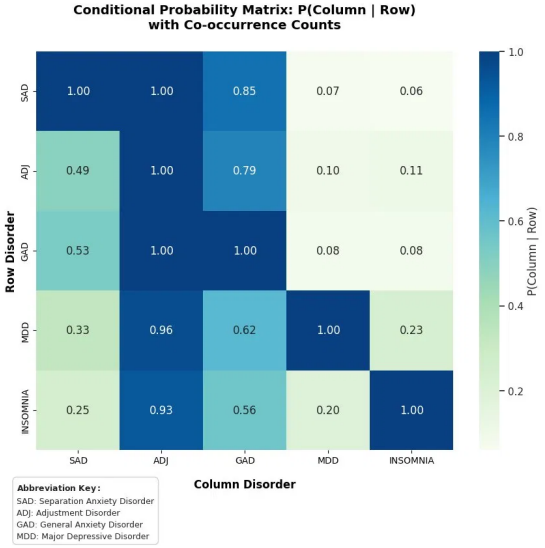


Figure 3: Conditional probability matrix $P(\text{Column Disorder} \mid \text{Row Disorder})$ for psychiatric symptom categories in RSPC ($n = 1{,}799$). Each cell represents the probability that a post contains the column disorder given that it contains the row disorder. Diagonal entries are 1.00 by definition. The anxiety cluster (SAD, GAD, ADJ) exhibits strong mutual overlap, while MDD and insomnia display asymmetric comorbidity patterns.

6.3 Relationships between Psychiatric Conditions and Relational Triggers

Figure 4 presents disorder-trigger co-occurrences with Ward linkage hierarchical clustering, showing similar trigger profiles for GAD, SAD, and ADJ, but a distinct profile for Insomnia. Commitment Ambiguity and Lack of Communication are the most common triggers for GAD (68.5%), SAD (79.0%), and ADJ (69.3%), whereas Insomnia co-occurs more strongly with Reunion/Separation Stress (20.3%), Silence Gap (14.1%), and Timezone Mismatch (6.2%). Consistent with shared mechanisms of anxious rumination and persistent worry across anxiety and insomnia (Harvey, 2002; Borkovec, 1994), timezone misalignment may disrupt circadian regulation and synchronous interaction, increasing communication scheduling friction in LDRs.

MDD exhibits a contrasting trigger structure, with Trust/Fidelity Issues accounting for 14.5% of triggers compared to 7.3–10.5% across GAD, SAD, and ADJ. This pattern aligns with cognitive and hopelessness-based models of depression, emphasizing negative appraisals and maladaptive attributions (Beck et al., 2024; Abramson et al., 1989). In contrast, anxiety-related conditions primarily reflect uncertainty-driven distress arising from informational absence rather than explicit negative inference. This distinction between appraisal-driven and uncertainty-driven distress parallels established psychopathology frameworks (Clark and Watson,

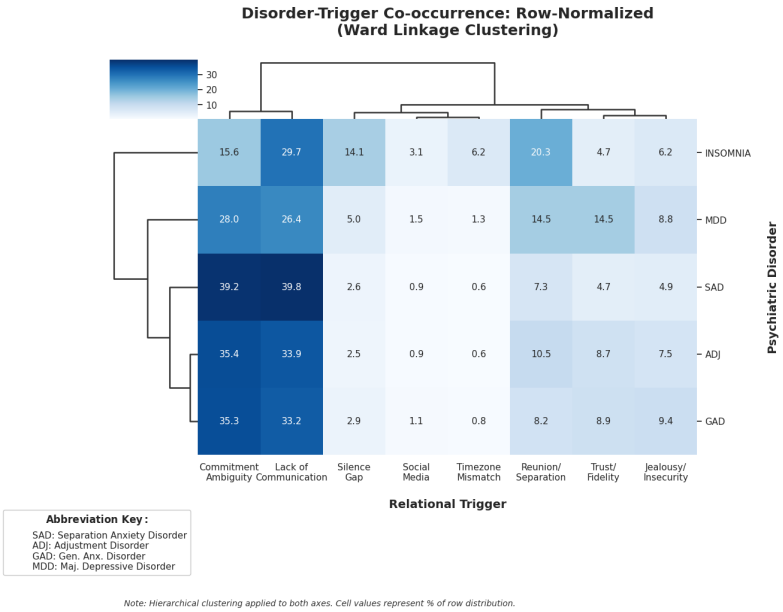


Figure 4: Row-normalized disorder–trigger co-occurrence heatmap with Ward linkage hierarchical clustering on the disorder axis. Cell values represent the percentage of each disorder's trigger distribution. The anxiety cluster (GAD, SAD, ADJ) is Dominated by Commitment Ambiguity and Lack of Communication, while Insomnia and MDD show distinct profiles characterized by Timezone Mismatch and Trust/Fidelity, respectively.

1991; Mineka et al., 2013), with RSPC providing computational evidence that can be recovered from naturalistic relational discourse.

6.4 Temporal and Event-Contingent Reasoning

Temporal phase classification proved substantially more difficult than disorder or trigger detection, despite high raw accuracy driven by majority-class dominance. Most transformer models collapsed toward the dominant SEPARATION phase (65.1% of instances), yielding substantially lower Macro-F1 scores. These findings suggest that event-contingent relational time is difficult to infer from isolated narratives and likely requires architectures capable of modeling sequential or state-transition dynamics, rather than standard prompting or classification approaches. Unlike psychiatric symptoms or relational triggers, temporal relationship states are often expressed indirectly through reunion planning, recency of travel, or anticipated separation, highlighting current NLP limitations in capturing latent relational timelines from narrative text alone.

7 Conclusion

This paper introduced the Relational Stress and Psychiatry Corpus (RSPC), the first clinically grounded benchmark for modeling psychiatric symptom expression in digitally mediated long-distance relationships. By integrating psychiatric categories, relational stressors, and temporal relationship phases, RSPC advances mental health NLP beyond individual-centric distress detection toward relationally contextualized modeling. Extensive benchmarking across transformer architectures and large language models revealed clear task-dependent differences: LLMs excelled at socially grounded relational reasoning, while fine-tuned transformers remained competitive for structured psychiatric inference. Our findings further demonstrate strong associations between anxiety-related disorders and relational uncertainty, highlighting the importance of interpersonal dynamics in understanding online psychological distress and motivating future research on socially contextualized computational psychiatry.

Limitations and Future Work

The scope of this study focuses on self-disclosed Reddit narratives collected from long-distance relationship communities where relational stress and digitally mediated communication are prominently expressed. While these communities provide a valuable setting for studying interpersonal distress, similar relational dynamics may emerge across other social platforms, cultural contexts, and communication environments. Expanding the benchmark to include broader demographic, multilingual,

and cross-platform settings could further enhance the diversity of relational expressions captured and support the development of more robust, generalizable relational mental health models. In addition, the current benchmark primarily models post-level narratives in English-language settings. Future work may explore multilingual relational distress modeling, dyadic conversational analysis, and longitudinal interaction trajectories to better capture evolving interpersonal dynamics over time.

The constrained classification performance partly stems from the temporal sparsity of social media narratives. While our annotations align with DSM-5-TR and ICD-11, these frameworks inherently assess pervasive, longitudinal patterns of symptoms. In contrast, isolated Reddit posts typically capture acute, episodic relational distress. Consequently, our labels represent textually inferred symptom expressions rather than formal clinical diagnoses. The inherent difficulty of extrapolating chronic psychiatric severity from cross-sectional snapshots of single relational stressors likely contributes to the performance ceilings observed across all evaluated architectures.

Data Adequacy and Validity: A critical limitation of the RSPC benchmark relates to data adequacy and the clinical validity of inferring complex psychiatric conditions from brief, temporally isolated text. Although the annotation framework is strictly aligned with DSM-5-TR and ICD-11 diagnostic criteria, formal clinical diagnoses inherently require the assessment of pervasive, persistent, and longitudinal behavioral patterns. In contrast, Reddit posts represent singular, often emotionally heightened snapshots of acute relational distress. Therefore, the diagnostic labels assigned in this corpus reflect episodic symptom presentations rather than comprehensive clinical histories, meaning the dataset cannot fully capture the chronic severity or pervasive nature of true psychiatric disorders.

Selection Biases: Furthermore, the generalizability of this benchmark is constrained by the demographic, cultural, and platform-specific selection biases inherent to our data collection. By sourcing data exclusively from English-language Reddit communities (r/LongDistance, r/LDR), the dataset disproportionately reflects the relationship norms, digital communication behaviors, and attachment paradigms of a younger, highly online, and predominantly Western demographic. This English-only, platform-specific approach fails to capture culturally diverse relationship dynamics, offline relational experiences, and non-English expressions of psychological distress. Consequently, models trained on this corpus risk underrepresenting older populations and non-Western cultural contexts, necessitating caution when applying these findings beyond digital, English-speaking ecosystems.

Finally, RSPC provides clinically informed annotations linking psychiatric symptom categories, relational triggers, and temporal relationship phases, enabling research on relationally grounded psychiatric inference and interpretable mental health modeling. Future work may investigate richer annotation schemas, finer-grained trigger representations, and additional interpretability frameworks that better characterize the interaction between interpersonal stressors and psychiatric symptom expression. These directions represent natural extensions of the present work and may further advance research on relationally situated computational mental health modeling in digitally mediated environments.

Ethical Statement

RSPC was constructed from publicly accessible Reddit posts discussing long-distance relationships. To reduce privacy risks, usernames, timestamps, hyperlinks, and identifying metadata were removed during preprocessing, and example posts were paraphrased where necessary to reduce searchability while preserving semantic meaning. Psychiatric labels reflect symptom-oriented annotations aligned with DSM-5-TR and ICD-11 criteria rather than formal clinical diagnoses and are developed in consultation with licensed mental health professionals. Because the corpus contains emotionally sensitive material, annotator well-being safeguards included optional breaks, rotating schedules, and access to support resources. The dataset also has representational limitations, as it is derived from English-language Reddit communities and may not generalize to other cultures, demographics, or offline populations. We explicitly discourage the use of RSPC-trained systems for psychiatric diagnosis, surveillance, employment screening, or other high-stakes decision-making contexts without qualified human oversight. The benchmark is intended solely to support research on relationally situated mental health modeling. Additional ethical safeguards, dataset governance procedures, and release considerations are provided in Appendix A.

This project did not require formal institutional

ethical review or an IRB protocol because the methodology did not constitute human subjects research under prevailing institutional guidelines. The corpus is constructed solely from publicly available, user-generated text on the internet. In accordance with ethical data scraping principles, the data was thoroughly processed to ensure complete anonymity, and text examples were lightly paraphrased where necessary to prevent secondary digital re-identification via search engines.

We acknowledge the use of large language models (LLMs), including generative AI tools, to assist with code development, grammatical refinement, formatting, and the structural organization and clarity of the manuscript. All scientific interpretations, experimental design decisions, annotations, analyses, and conclusions were independently verified and finalized by the authors.

References

Lyn Y Abramson, Gerald I Metalsky, and Lauren B Alloy. 1989. Hopelessness depression: A theory-based subtype of depression. *Psychological review*, 96(2):358.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, and 1 others. 2024. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*.

Sharifa Alghowinem, Roland Goecke, Michael Wagner, Julien Epps, Matthew Hyett, Gordon Parker, and Michael Breakspear. 2016. Multimodal depression detection: fusion analysis of paralinguistic, head pose and eye gaze behaviors. *IEEE Transactions on Affective Computing*, 9(4):478–490.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the 2nd clinical natural language processing workshop*, pages 72–78.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic.

American Psychiatric Association. 2022. *Diagnostic and statistical manual of mental disorders*. American Psychiatric Association Publishing.

Aaron T Beck, A John Rush, Brian F Shaw, Gary Emery, Robert J DeRubeis, and Steven D Hollon. 2024. *Cognitive therapy of depression*. Guilford Publications.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Debarun Bhattacharjya, Balaji Ganesan, Junkyu Lee, Radu Marinescu, Katsiaryna Mirylenka, Michael Glass, and Xiao Shou. 2025. Simba uq: Similarity-based aggregation for uncertainty quantification in large language models. *arXiv preprint arXiv:2510.13836*.

Thomas D Borkovec. 1994. The nature, functions, and origins of worry.

R Nicholas Carleton. 2016. Into the unknown: A review and synthesis of contemporary models involving uncertainty. *Journal of anxiety disorders*, 39:30–43.

Stevie Chancellor, Michael L Birnbaum, Eric D Caine, Vincent MB Silenzio, and Munmun De Choudhury. 2019. A taxonomy of ethical tensions in inferring mental health states from social media. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 79–88.

Siyuan Chen, Meilin Wang, Minghao Lv, Zhiling Zhang, Juqianqian Juqianqian, Dejiyangla Dejiyangla, Yujia Peng, Kenny Zhu, and Mengyue Wu. 2024. Mapping long-term causalities in psychiatric symptomatology and life events from social media. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5472–5487.

Lee Anna Clark and David Watson. 1991. Tripartite model of anxiety and depression: psychometric evidence and taxonomic implications. *Journal of abnormal psychology*, 100(3):316.

Arman Cohan, Bart Desmet, Andrew Yates, Luca Soldaini, Sean MacAvaney, and Nazli Goharian. 2018. Smhd: a large-scale resource for exploring online language usage for multiple mental health conditions. In *Proceedings of the 27th international conference on computational linguistics*, pages 1485–1497.

Glen Coppersmith, Mark Dredze, Craig Harman, Kristy Hollingshead, and Margaret Mitchell. 2015. Clpsych 2015 shared task: Depression and ptsd on twitter. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality*, pages 31–39.

Munmun De Choudhury, Michael Gamon, Scott Counts, and Eric Horvitz. 2013. Predicting depression via social media. In *Proceedings of the international AAAI conference on web and social media*, volume 7, pages 128–137.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*, pages 4171–4186.

Michel J Dugas, Fabien Gagnon, Robert Ladouceur, and Mark H Freeston. 1998. Generalized anxiety disorder: A preliminary test of a conceptual model. *Behaviour research and therapy*, 36(2):215–226.

Colin A Espie. 2007. Understanding insomnia through cognitive modelling. *Sleep Medicine*, 8:S3–S8.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Zhijun Guo, Alvina Lai, Johan H Thygesen, Joseph Farrington, Thomas Keen, and Kezhi Li. 2024. Large language models for mental health applications: systematic review. *JMIR mental health*, 11(1):e57400.

Keith Harrigian, Carlos Aguirre, and Mark Dredze. 2021. On the state of social media data for mental health research. In *Proceedings of the Seventh Workshop on Computational Linguistics and Clinical Psychology: Improving Access*, pages 15–24.

Allison G Harvey. 2002. A cognitive model of insomnia. *Behaviour research and therapy*, 40(8):869–893.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7871–7880.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

David E Losada, Fabio Crestani, and Javier Parapar. 2018. Overview of erisk: early risk prediction on the internet. In *International conference of the cross-language evaluation forum for european languages*, pages 343–361. Springer.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Daniel M Low, Laurie Rumker, Tanya Talkar, John Torous, Guillermo Cecchi, and Satrajit S Ghosh. 2020. Natural language processing reveals vulnerable mental health support groups and heightened health anxiety on reddit during covid-19: Observational study. *Journal of medical Internet research*, 22(10):e22635.

Minghao Lv, Siyuan Chen, Haoan Jin, Minghao Yuan, Qianqian Ju, Yujia Peng, Kenny Zhu, and Mengyue Wu. 2025. Tracking life's ups and downs: Mining life events from social media posts for mental health analysis. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6950–6965.

Sean MacAvaney, Bart Desmet, Arman Cohan, Luca Soldaini, Andrew Yates, Ayah Zirikly, and Nazli Goharian. 2018. Rsdd-time: Temporal annotation of self-reported mental health diagnoses. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 168–173.

Susan Mineka, David Watson, and Lee Anna Clark. 2013. Comorbidity of anxiety and unipolar mood disorders. *Fear and anxiety*, pages 113–148.

Charles M Morin. 1993. *Insomnia: Psychological assessment and management.* Guilford press.

Moin Nadeem. 2016. Identifying depression on twitter. *arXiv preprint arXiv:1607.07384*.

Carman Neustaedter and Saul Greenberg. 2012. Intimacy in long-distance relationships over video chat. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 753–762.

Alina Pavlova and Pauwke Berkers. 2020. Mental health discourse and social media: Which mechanisms of cultural power drive discourse on twitter. *Social Science & Medicine*, 263:113250.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and 1 others. 2011. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Ivan Sekulić, Matej Gjurković, and Jan Šnajder. 2018. Not just depressed: Bipolar disorder prediction on reddit. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 72–78.

Laura Stafford. 2004. *Maintaining long-distance and cross-residential relationships*. Routledge.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Matthijs J. Warrens. 2008. On association coefficients for 2×2 tables and properties that do not depend on the marginal distributions. *Psychometrika*, 73:777–789.

JR World Health Organization and 1 others. 2018. International classification of diseases for mortality and morbidity statistics (11th revision).

Andrew Yates, Arman Cohan, and Nazli Goharian. 2017. Depression and self-harm risk assessment in online forums. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 2968–2978.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and 1 others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Ayah Zirikly, Philip Resnik, Ozlem Uzuner, and Kristy Hollingshead. 2019. Clpsych 2019 shared task: Predicting the degree of suicide risk in reddit posts. In *Proceedings of the sixth workshop on computational linguistics and clinical psychology*, pages 24–33.

Appendix-1: Inter-Annotator Agreement Scores

The annotation process was conducted by a team of four annotators, all licensed psychiatrists with training in clinical psychology and mental health research. Because the expression of psychiatric symptoms and relational distress in social media narratives can be subjective and context-dependent, detailed annotation guidelines were established prior to annotation to improve consistency among annotators.

Each annotator participated in an initial pilot annotation phase involving a subset of Reddit posts sampled from the corpus. Disagreements were reviewed collaboratively under the supervision from clinically informed adjudicators, and the annotation manual was iteratively refined to clarify category boundaries, reduce ambiguity, and standardize labeling criteria before full-scale annotation began.

Annotations were performed across three tiers:

- **Tier A: Psychiatric Symptom Categories.** Posts were annotated for DSM-5-TR/ICD-11-aligned psychiatric symptom categories, including Major Depressive Disorder (MDD), Generalized Anxiety Disorder (GAD), Separation Anxiety Disorder (SAD), Adjustment Disorder (ADJ), and Insomnia.
- **Tier B: Relational Stressor Triggers.** Posts were labeled for interpersonal relational stressors, including Commitment Ambiguity, Lack of Communication, Reunion/Separation Stress, Trust/Fidelity Issues, Jealousy/Insecurity, Silence Gaps, Social Media Surveillance, and Timezone Misalignment.
- **Tier C: Temporal Relationship Phases.** Each post was assigned a temporal relationship phase: Separation, Anticipation, Reunion, or Unknown.

To evaluate annotation reliability, we computed multiple inter-annotator agreement (IAA) metrics, including Cohen's κ, Fleiss' κ, and Krippendorff's α. Pairwise Cohen's κ values were computed across annotator pairs, while Fleiss' κ and Krippendorff's α were computed collectively across all annotators.

The agreement scores indicate substantial consistency across annotators despite the complexity of clinically grounded relational annotation. The average pairwise Cohen's κ score was 0.774, while the collective Krippendorff's α and Fleiss' κ scores were 0.761 and 0.747, respectively. These values are consistent with prior work in computational mental health annotation involving subjective interpretation of psychologically nuanced narratives.

Annotator Pair	Krippendorff's α	Cohen's κ	Fleiss' κ
(1,2)	0.781	0.794	-
(1,3)	0.753	0.768	-
(1,4)	0.737	0.748	-
(2,3)	0.769	0.782	-
(2,4)	0.721	0.736	-
(3,4)	0.803	0.816	-
All Annotators	0.761	-	0.747

Table 7: Inter-Annotator Agreement Scores Across Annotation Tiers

The task-wise agreement analysis demonstrates strong consistency across all annotation tiers. Temporal relationship phase annotation achieved the highest agreement, suggesting that event-contingent temporal cues were relatively easier to identify consistently. Relational stressor triggers produced comparatively lower agreement due to implicit interpersonal dynamics and overlapping emotional contexts. Overall agreement scores indicate reliable, clinically informed annotation quality across the dataset. For task-wise agreement analysis, Cohen's κ values represent the average pairwise agreement across all annotator pairs, while Krippendorff's α and Fleiss' κ were computed collectively across all four annotators.

Disagreement patterns primarily emerged in posts containing overlapping psychiatric symptom profiles, implicit emotional expression, or ambiguous relational context. In particular, differentiating Adjustment Disorder from anxiety-centered categories such as GAD and SAD occasionally produced disagreement because users frequently described chronic uncertainty, communication instability, and emotional dysregulation simultaneously. Similarly, implicit insomnia-related behaviors (e.g., staying awake awaiting messages, disrupted sleep schedules due to time zone differences) led to occasional variability in annotations due to indirect symptom expression.

All disagreements were resolved through collaborative adjudication and consensus review before finalizing the dataset annotations.

Annotation Tier / Task	Avg. Pairwise Cohen's κ	Krippendorff's α	Fleiss' κ	Agreement Level
Tier A: Psychiatric Symptom Categories	0.780	0.768	0.754	Substantial
Tier B: Relational Stressor Triggers	0.720	0.708	0.694	Substantial
Tier C: Temporal Relationship Phases	0.810	0.798	0.783	Almost Perfect
Overall Across All Tiers	**0.774**	**0.761**	**0.747**	**Substantial**

Table 8: Task-wise Inter-Annotator Agreement Scores Across Four Annotators

Appendix-2: RSPC Annotation Taxonomy

Figure 5 presents the complete RSPC annotation taxonomy, illustrating the three complementary annotation dimensions applied to each Reddit post in the corpus: (1) Psychiatric Diagnostic Categories, (2) Relational Stressor Triggers, and (3) Temporal Relationship Phase.

A. Annotation Principles

The taxonomy was designed around four core principles. First, annotations are **clinically grounded**, aligned with DSM-5-TR and ICD-11 diagnostic criteria developed in collaboration with licensed psychiatrists rather than lay annotators. Second, Tasks 1 and 2 support **multi-label** annotation, reflecting the clinical reality that multiple psychiatric symptoms and relational stressors frequently co-occur within a single post. Third, all labels are assigned at the **post level**, treating each Reddit narrative as the primary unit of analysis. Fourth, label definitions are **evidence-based**, operationalized using published diagnostic guidelines to ensure reproducibility and clinical validity.

B. Tier Descriptions

Task 1 — Psychiatric Diagnostic Categories (DSM-5-TR/ICD-11 Aligned). Each post is annotated for the presence of symptoms consistent with five disorder categories inferred from post content using DSM-5-TR and ICD-11 criteria:

- **SAD (Separation Anxiety Disorder):** Excessive fear or anxiety about separation from attachment figures, including distress during separation, nightmares, or somatic symptoms.
- **ADJ (Adjustment Disorder):** Emotional or behavioral symptoms arising in response to an identifiable stressor, occurring within three months of the stressor onset and disproportionate to its severity.
- **GAD (Generalized Anxiety Disorder):** Excessive, difficult-to-control worry across multiple life domains, accompanied by restlessness, fatigue, or physical tension.
- **MDD (Major Depressive Disorder):** Persistent low mood or anhedonia lasting more than two weeks, with associated feelings of worthlessness, hopelessness, or psychomotor agitation.
- **Insomnia Disorder:** Dissatisfaction with sleep quality or quantity (difficulty initiating or maintaining sleep) causing clinically significant distress.

Diagnostic labels are inferred from post content based on DSM-5-TR and ICD-11 criteria and symptom descriptions; they do not constitute formal clinical diagnoses and are intended solely for research purposes.

Task 2 — Relational Stressor Triggers. Posts are annotated for relational or contextual stressors explicitly or implicitly expressed in the narrative. Eight trigger categories are defined:

- **Commitment Ambiguity:** Uncertainty about the status, future, or seriousness of the relationship.
- **Lack of Communication:** Persistent or recurring difficulty communicating with the partner.
- **Reunion/Separation Stress:** Emotional strain surrounding periods of physical reunion or separation.
- **Trust/Fidelity Issues:** Concerns about a partner's exclusivity, honesty, or reliability.
- **Jealousy/Insecurity:** Expressions of worry, mistrust, or perceived threat arising from third-party interactions.
- **Silence Gap:** Unexpectedly reduced or absent communication, including being left on read or ghosted.

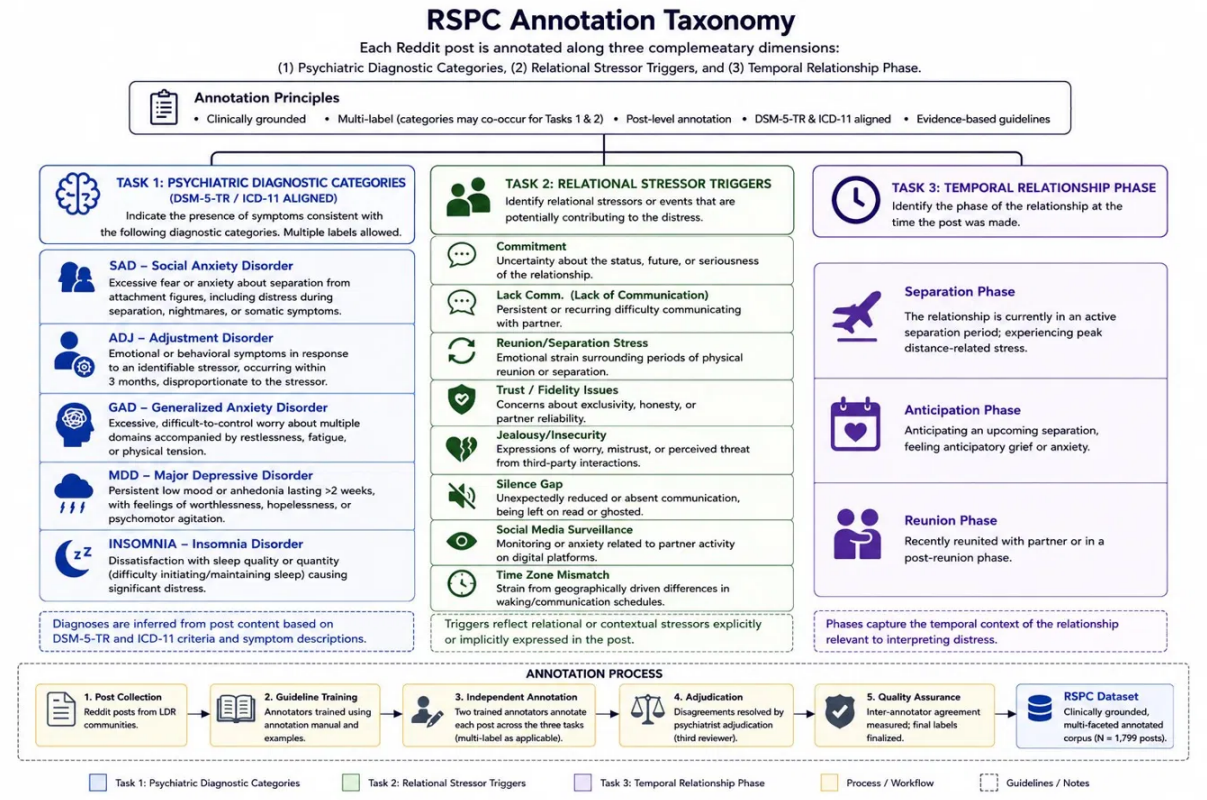


Figure 5: RSPC Annotation Taxonomy. Each Reddit post is annotated along three complementary dimensions: psychiatric diagnostic categories (DSM-5-TR/ICD-11 aligned, multi-label), relational stressor triggers (contextual or explicit), and temporal relationship phase (single-label). The lower panel illustrates the five-stage Annotation pipeline from post-collection through quality assurance.

- **Social Media Surveillance:** Monitoring of or anxiety related to a partner's activity on digital platforms.
- **Timezone Misalignment:** Strain arising from geographically driven differences in waking hours or communication schedules.

Task 3 — Temporal Relationship Phase. Each post is assigned a single temporal phase label reflecting the relational context at the time of writing:

- **Separation:** The relationship is in an active separation period, with the author experiencing peak stress related to distance.
- **Anticipation:** The author is anticipating an upcoming reunion or separation, experiencing anticipatory grief or anxiety.
- **Reunion:** The author has recently reunited with their partner or is in a post-reunion phase.
- **Unknown:** Insufficient temporal information to assign a definitive phase label.

Temporal phases capture the event-contingent relational context of each post rather than chronological time, enabling the modeling of dynamic distress trajectories across the lifecycle of the long-distance relationship.

C. Annotation Pipeline

The annotation workflow proceeded through five sequential stages, as illustrated in the lower panel of Figure 5:

1. **Post Collection.** Reddit posts were collected from long-distance relationship communities (`r/LongDistance`, `r/LDR`) and filtered for narrative completeness, relational relevance, and linguistic consistency.
2. **Guideline Training.** Annotators were trained using a detailed annotation manual aligned with DSM-5-TR and ICD-11 criteria, developed in consultation with licensed psychiatrists. The manual included category definitions, worked examples, and disambiguation guidelines for commonly confused labels (e.g.,

ADJ vs. GAD, Silence Gap vs. Lack of Communication).

3. **Independent Annotation.** Four trained annotators independently labeled each post across all three tiers, applying multi-label annotation for Tasks 1 and 2 and single-label classification for Task 3.

4. **Adjudication.** Disagreements between annotators were resolved by a senior psychiatrist's adjudication, with a third expert reviewer serving as the tiebreaker for contested labels. Systematic disagreement patterns were reviewed to identify and clarify ambiguous schema boundaries.

5. **Quality Assurance.** Inter-annotator agreement was measured using Cohen's κ, Fleiss' κ, and Krippendorff's α across all annotation tiers (see Appendix-1). Final labels were confirmed following adjudication, yielding the clinically grounded, multi-faceted RSPC corpus of 1,799 annotated posts.

A Appendix-3: Advanced Prompting and Uncertainty Quantification

To overcome limitations of deterministic prompting, an extensive ablation study evaluated three diagnostic tasks using progressively enhanced prompting strategies. The standard three-shot, zero-temperature baseline was augmented with structured Chain-of-Thought reasoning, requiring models to explicitly connect narrative evidence with diagnostic criteria before making a prediction. Clinically tailored instructions were then incorporated to strengthen contextual understanding and clinical reasoning. These controlled prompt refinements isolated improvements attributable solely to prompt engineering before introducing more sophisticated decoding and uncertainty estimation strategies into the evaluation pipeline.

After optimizing prompts, deterministic decoding was replaced by the Similarity-Based Aggregation for Uncertainty Quantification (Simba-UQ) framework (Bhattacharjya et al., 2025). Instead of producing a single prediction, multiple reasoning paths were generated using temperature sampling at progressively higher temperatures. Predicted label sets from these outputs were compared using Jaccard similarity, and arithmetic aggregation was used to select the most consistent clinical inference. This strategy enabled robust uncertainty estimation while simultaneously improving calibration, confidence assessment, and predictive performance across diverse diagnostic classification tasks under systematic evaluation.

Experimental results demonstrate that prompt engineering consistently improves diagnostic performance across all evaluated large language models. While clinically tailored instructions combined with Chain-of-Thought reasoning produced modest gains over deterministic prompting, integrating Simba-UQ yielded the highest macro-F1 and AUC, as well as a substantially lower Adaptive Calibrated Error, indicating more reliable confidence estimates. These findings establish that combining advanced reasoning with uncertainty quantification yields accurate, well-calibrated predictions, thereby addressing critical trust requirements for the responsible deployment of large language models in sensitive automated healthcare environments.

Across all three tasks (Tables 9–11), progressively stronger prompting consistently improves predictive performance, with Chain-of-Thought reasoning and clinically tailored instructions increasing Macro-F1, Micro-F1, Weighted-F1, AUC, and accuracy where applicable. Integrating Simba-UQ further achieves the highest classification performance while minimizing Adaptive Calibrated Error (ACE), demonstrating improved calibration and confidence reliability. LLaMA-3-70B generally performs best for multi-label classification and temporal phase prediction, GPT-4o leads in relational trigger detection, and Claude-3-Haiku exhibits consistent relative gains, confirming the framework's robustness across language models of varying capacities.

Appendix-4: Dataset Examples

Table 12 presents three representative posts from RSPC illustrating the diversity of psychiatric symptom expressions, relational trigger profiles, and temporal phases captured in the corpus.

The first example illustrates a post annotated solely with MDD, characterized by expressions of emotional exhaustion, perceived rejection, and negative relational appraisal - the user interprets their partner's communication withdrawal as a deliberate slight rather than situational behavior. The trigger profile indicates Lack of Communication as the primary stressor, consistent with the MDD — Trust/Fidelity association observed in the row-

Table 9: Performance of Large Language Models on Task 1 (Multi-Label Disorder Classification) under advanced prompting strategies and similarity-based aggregation. The table contrasts the baseline deterministic 0-temperature decoding with Chain-of-Thought (CoT) and clinically tailored instructions, further evaluating the Simba-UQ temperature sampling approach to measure both classification performance (F1, AUC) and confidence calibration (Adaptive Calibrated Error).

Model	Prompting Strategy	Macro-F1	Micro-F1	Wt.-F1	AUC	ACE
Claude-3-Haiku	3-shot (Base)	0.483	0.567	0.566	0.592	–
	3-shot + CoT	0.497	0.579	0.580	0.606	0.089
	Clinical Prompt	0.517	0.597	0.600	0.624	0.066
	Clinical Prompt + Simba-UQ	**0.526**	**0.603**	**0.607**	**0.633**	**0.039**
GPT-4o	3-shot (Base)	0.438	0.498	0.521	0.613	–
	3-shot + CoT	0.456	0.514	0.536	0.626	0.102
	Clinical Prompt	0.479	0.536	0.558	0.644	0.073
	Clinical Prompt + Simba-UQ	**0.491**	**0.547**	**0.569**	**0.655**	**0.044**
LLaMA-3-70B	3-shot (Base)	0.503	0.596	0.582	0.622	–
	3-shot + CoT	0.514	0.608	0.594	0.635	0.086
	Clinical Prompt	0.529	0.619	0.607	0.648	0.061
	Clinical Prompt + Simba-UQ	**0.536**	**0.626**	**0.615**	**0.656**	**0.036**

Table 10: Performance of Large Language Models on Task 2 (Relational Trigger Detection) under progressively enhanced prompting strategies and similarity-based uncertainty aggregation. Starting from deterministic 3-shot prompting, we evaluate the impact of Chain-of-Thought (CoT), clinically tailored prompting, and the Simba-UQ multi-temperature aggregation framework on predictive performance and calibration.

Model	Prompting Strategy	Macro-F1	Micro-F1	Wt.-F1	AUC	ACE
GPT-4o	3-shot (Base)	0.519	0.576	0.648	0.743	–
	3-shot + CoT	0.533	0.589	0.660	0.754	0.091
	Clinical Prompt	0.548	0.603	0.673	0.767	0.063
	Clinical Prompt + Simba-UQ	**0.558**	**0.612**	**0.682**	**0.776**	**0.037**
LLaMA-3-70B	3-shot (Base)	0.492	0.567	0.587	0.736	–
	3-shot + CoT	0.505	0.579	0.599	0.747	0.088
	Clinical Prompt	0.519	0.592	0.612	0.759	0.061
	Clinical Prompt + Simba-UQ	**0.528**	**0.600**	**0.620**	**0.768**	**0.035**
Claude-3-Haiku	3-shot (Base)	0.483	0.597	0.534	0.571	–
	3-shot + CoT	0.496	0.607	0.548	0.585	0.095
	Clinical Prompt	0.512	0.619	0.563	0.601	0.069
	Clinical Prompt + Simba-UQ	**0.521**	**0.628**	**0.572**	**0.612**	**0.041**

Table 11: Performance of Large Language Models on Task 3 (Temporal Phase Classification) under advanced prompting and Simba-UQ configurations. The evaluation includes accuracy and F1 metrics for event-contingent phase prediction, supplemented by Adaptive Calibrated Error (ACE) to quantify the reliability of the models' temporal reasoning.

Model	Prompting Strategy	Accuracy (↑)	Macro-F1 (↑)	Micro-F1 (↑)	Wt.-F1 (↑)	ACE (↓)
LLaMA-3-70B	3-Shot (Base)	0.665	0.374	0.665	0.613	–
	3-Shot + CoT	0.674	0.386	0.674	0.622	–
	Clinically Tailored	0.686	0.401	0.686	0.635	–
	Clinically Tailored + Simba-UQ	**0.691**	**0.408**	**0.691**	**0.641**	**0.047**
GPT-4o	3-Shot (Base)	0.601	0.291	0.601	0.563	–
	3-Shot + CoT	0.614	0.307	0.614	0.575	–
	Clinically Tailored	0.626	0.323	0.626	0.588	–
	Clinically Tailored + Simba-UQ	**0.632**	**0.331**	**0.632**	**0.595**	**0.061**
Claude-3-Haiku	3-Shot (Base)	0.485	0.294	0.485	0.537	–
	3-Shot + CoT	0.503	0.311	0.503	0.551	–
	Clinically Tailored	0.519	0.329	0.519	0.566	–
	Clinically Tailored + Simba-UQ	**0.526**	**0.337**	**0.526**	**0.572**	**0.058**

Table 12: Representative RSPC examples illustrating annotation across psychiatric symptom categories, relational stressor triggers, and temporal relationship phases. Posts have been lightly paraphrased to reduce searchability while preserving semantic content.

Post (Anonymized)	**Disorders**	**Triggers**	**Phase**
"She already emotionally detached two or three months ago. She said 'Well they actually are here with me to support me, of course I gotta put them first.' Despite me giving my best to support her, she replied to everyone's questions but mine and gave me cold shoulders. So to all out there, don't do this to your LDRs."	MDD	Lack of Communication	Separation
"I'm having troubles trusting my partner whenever I try to check his conversations with his girl friends. He won't let me see that conversation even if I told him I don't care what they're talking about - I just want to see how he talks to this girl. How can I trust him if he's not willing to be open about it? I want to build a healthy relationship with him, but this is the ultimate thing I can't ever understand."	ADJ, GAD, SAD	Trust/Fidelity Issues	Separation
"I love being with him and calling him, but a lot of the time we end up sitting in silence or just trading the usual 'I love you' and 'I miss you' back and forth. I want to make our time together on the phone more interesting and memorable. Do you have any suggestions on things we can do?"	ADJ, GAD, SAD	Commitment Ambiguity, Lack of Communication	Separation

normalized heatmap analysis.

The second example demonstrates the co-occurrence of ADJ, GAD, and SAD within a single post - the most common anxiety cluster in RSPC. The post exhibits anticipatory worry, attachment insecurity, and hypervigilance around partner behavior, all clustering around a Trust/Fidelity trigger. Notably, the user oscillates between catastrophizing ("how can I trust him") and reassurance-seeking ("I know they're just friends"), a pattern consistent with the intolerance-of-uncertainty mechanism in GAD.

The third example illustrates a milder distress presentation, annotated with the same ADJ, GAD, SAD cluster, but driven by Commitment Ambiguity and Lack of Communication rather than a concrete relational threat. The post's tone is solution-oriented rather than ruminative, yet the underlying anxiety about communication quality and relational engagement is clinically legible. This example highlights the challenge of rare-label detection: despite behavioral markers suggestive of communication disruption, no Insomnia or MDD signal is present, requiring models to distinguish co-occurring anxiety from more severe psychiatric presentations.

Table 13: Task 1 Per-Label Results (BERT-base vs. GPT-4o few-shot).

	BERT-base		**GPT-4o**	
Label	**Prec.**	**F1**	**Prec.**	**F1**
SAD	0.498	0.581	0.594	0.361
ADJ	0.753	0.808	0.794	0.703
GAD	0.738	0.759	0.923	0.312
MDD	0.273	0.370	0.523	0.438
Insomnia	0.002	0.003	0.600	0.375

Appendix-5: Per-Label Classification Results for Task-1

Per-label evaluation in Table 13 provides a finer-grained interpretation of **Task 1** outcomes with respect to **RQ1**. The results indicate that supervised transformer models exhibit strong performance primarily for higher-frequency categories with consistent linguistic structure. BERT-base achieves the highest F1 scores for ADJ (0.808), GAD (0.759), and SAD (0.581), suggesting effective representation learning when clinically relevant lexical and semantic patterns are sufficiently represented in the training. However, performance deteriorates substantially for MDD (0.370) and Insomnia (0.003).

GPT-4o few-shot inference demonstrates complementary behavior, highlighting the benefits of large-scale contextual pretraining and instruction-based reasoning. Although overall F1 performance remains lower for SAD (0.361) and GAD (0.312),

GPT-4o achieves substantially higher precision across all categories, including GAD (0.923) and ADJ (0.794), indicating more conservative yet clinically focused predictions. Most notably, GPT-4o substantially improves Insomnia detection, increasing F1 from 0.003 to 0.375, while also outperforming BERT-base on MDD. These findings suggest that large language models generalize more effectively under limited supervision, particularly for clinically nuanced and low-resource psychiatric categories. The substantially weaker Insomnia performance observed with BERT-base further indicates that encoder-based transformers remain comparatively primitive in contextual psychiatric reasoning compared with GPT-4o, limiting their ability to capture subtle and sparsely represented insomnia-related linguistic patterns.

Appendix-6: Statistical Rigor and Reproducibility

A. Experimental Reproducibility

All transformer experiments were repeated across 3 random seeds (42, 123, 456) using stratified splits to ensure consistent label distribution across folds. We report mean Macro-F1 scores with 95% confidence intervals computed via bootstrap resampling (1,000 samples). Table 14 presents detailed reproducibility statistics for the top-performing models on each task.

B. Statistical Significance Testing

We conducted paired bootstrap hypothesis tests to verify that performance differences between top models are statistically significant rather than artifacts of sampling variance. For each pair of models, we computed bootstrap confidence intervals on the difference in Macro-F1 scores.

Table 15 reports p-values from two-sided paired t-tests comparing top-performing models within each architecture family (transformers vs. LLMs) and across families. Significance levels: $*$ ($p < 0.05$), $**$ ($p < 0.01$), $***$ ($p < 0.001$).

Appendix-7: Hyperparameter Specifications

Table 16 provides complete hyperparameter configurations for all fine-tuned transformer models. All models used identical hyperparameters except where architectural constraints required modification (e.g., Longformer and BigBird use longer maximum sequence lengths to exploit their efficient attention mechanisms).

A. Optimizer Configuration.

All models used AdamW (Loshchilov and Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1e-8$. Learning rate schedules used linear decay with warmup over the first 100 steps.

B. Early Stopping.

Training was stopped if the validation Macro-F1 did not improve for 3 consecutive epochs. Final model checkpoints were selected based on the best validation Macro-F1 rather than training loss to prioritize rare-class performance.

C. Class Weighting.

For multi-label tasks, we applied inverse-frequency weighting: $w_c = \frac{N}{2 \times n_c}$, where N is the total number of training instances and n_c is the number of positive instances for class c. Class weights were capped at $w_{\max} = 10.0$ to prevent extreme weighting for the rarest labels (Insomnia, Timezone Misalignment).

Appendix-8: Cross-Validation Results for Temporal Phase Classification

We conducted a 3-fold cross-validation on the full RSPC dataset using logistic regression as a controlled baseline classifier. Three configurations were evaluated: an unweighted baseline, class reweighting using inverse-frequency weights, and random oversampling of minority phases. Macro-F1 was used as the primary metric to emphasize performance on minority classes. Table 17 reports fold-level scores and mean $\pm$ standard deviation across folds.

Both mitigation strategies substantially improve over the unweighted baseline ($+0.179$ Macro-F1 for class re-weighting; $+0.183$ for oversampling). Class re-weighting achieves a marginally higher mean Macro-F1 (0.4064) but exhibits greater variance across folds (± 0.0147) relative to oversampling (± 0.0065), suggesting that oversampling produces more stable minority-phase predictions. Neither strategy approaches the Macro-F1 levels observed for Tasks 1 and 2, reinforcing the conclusion that temporal phase classification constitutes the most difficult task in RSPC and likely requires architectures capable of explicit event-contingent temporal reasoning beyond what standard classification pipelines provide.

Table 14: Reproducibility Statistics: Mean performance across 3 random seeds with 95% confidence intervals (bootstrap, n=1,000).

Task	Model	Mean Macro-F1	95% CI	Std. Dev.
Task 1	BERT-base	0.5035	[0.4891, 0.5179]	0.0073
	BigBird-RoBERTa	0.5050	[0.4912, 0.5188]	0.0071
	Claude-3-Haiku	0.5376	[0.5201, 0.5551]	0.0089
Task 2	BigBird-RoBERTa	0.4779	[0.4623, 0.4935]	0.0079
	LLaMA-3-70B	0.4919	[0.4751, 0.5087]	0.0086
	GPT-4o	0.5189	[0.5012, 0.5366]	0.0090
Task 3	BERT-base	0.5630	[0.5201, 0.6059]	0.0219
	T5-base	0.5630	[0.5189, 0.6071]	0.0225

Table 15: Statistical Significance: Pairwise comparisons between top models using paired bootstrap tests.

Task	Comparison	Δ Macro-F1	p-value	Sig.
Task 1	Claude-3 vs. BigBird	+0.0326	0.0078	**
	Claude-3 vs. BERT	+0.0341	0.0065	**
	BigBird vs. BERT	+0.0015	0.8234	n.s.
Task 2	GPT-4o vs. BigBird	+0.0410	0.0003	***
	GPT-4o vs. LLaMA-3	+0.0270	0.0189	*
	LLaMA-3 vs. BigBird	+0.0140	0.1423	n.s.
Task 3	BERT/T5 vs. Majority	+0.1679	<0.0001	***

Appendix-9: Large Language Model Prompting Details & Prompt Templates

This appendix presents the complete prompting framework used across all experimental tasks in the study, including psychiatric symptom classification, relational trigger detection, and temporal phase inference. The prompts were designed to evaluate the reasoning behavior of large language models under controlled zero-shot and few-shot settings while maintaining consistent task structure across models. Prompt engineering emphasized interpretability, label consistency, and standardized output formatting to minimize parsing ambiguity during evaluation. Clinical and relational labels were operationalized using concise, DSM-5-TR-aligned descriptions and psychologically grounded definitions, enabling models to infer latent emotional states from naturalistic Reddit narratives without requiring additional contextual metadata.

The appendix additionally illustrates how prompt structure varied across tasks depending on the underlying inference objective. Multi-label psychiatric classification required clinically descriptive instructions and explicit symptom definitions, whereas relational trigger detection relied more heavily on contextual examples to improve semantic grounding. Temporal phase classification employed concise zero-shot formulations to assess whether models could infer implicit relational states directly from narrative cues. Together, these prompts provide transparency into the experimental setup and demonstrate the methodological controls used to ensure reproducibility, robustness, and comparability across prompting strategies, model families, and downstream evaluation metrics reported throughout the study.

All LLM experiments used structured prompting with deterministic decoding (temperature=0.0) to ensure reproducibility. Below are the exact prompt templates used for each task.

A. Task 1: Multi-Label Disorder Classification

A.1 Zero-Shot Prompt

Classification Prompt (Zero-Shot)

You are a clinical psychology expert. Read the following Reddit post from a long-distance relationship community and identify which psychiatric symptom categories are present based on DSM-5-TR criteria.

Table 16: Complete Hyperparameter Specifications for Fine-Tuned Transformers.

Model	Max Len	Batch Size	LR	Epochs	Weight Decay	Warmup Steps
BERT-base	512	16	2e-5	10	0.01	100
RoBERTa-base	512	16	2e-5	10	0.01	100
ClinicalBERT	512	16	2e-5	10	0.01	100
BART-base	512	16	2e-5	10	0.01	100
T5-base	512	16	2e-5	10	0.01	100
Longformer	1024	8	2e-5	10	0.01	100
BigBird-RoBERTa	1024	8	2e-5	10	0.01	100

Table 17: 3-Fold Cross-Validation Macro-F1 results for Task 3 (Temporal Phase Classification) on the full RSPC dataset. Mean and standard deviation are computed across three folds.

Mitigation Scheme	Fold 1	Fold 2	Fold 3	Mean $\pm$ SD
Baseline (Unweighted LR)	0.2319	0.2323	0.2173	0.2271 ± 0.0077
Class Re-weighted LR	0.4239	0.3957	0.3995	0.4064 ± 0.0147
Oversampling LR	0.4032	0.4163	0.4101	0.4099 ± 0.0065

Possible Labels

- **SAD**: Separation Anxiety Disorder (excessive anxiety about separation from attachment figures)
- **ADJ**: Adjustment Disorder (emotional/behavioral symptoms in response to identifiable stressors)
- **GAD**: Generalized Anxiety Disorder (excessive worry across multiple domains)
- **MDD**: Major Depressive Disorder (persistent low mood, anhedonia, hopelessness)
- **Insomnia**: Sleep disturbance (difficulty falling or staying asleep)

Post

`{POST_TEXT}`

Return **ONLY** a comma-separated list of applicable labels (for example: `SAD, ADJ, GAD`) or `None` if no symptoms are evident.

Labels:

A.2 Few-Shot Prompt

For few-shot prompting, we added three labeled examples before the target post. Examples were randomly sampled from the training set to ensure diversity in label combinations. Example format:

Classification Prompt (Few-Shot)

You are a clinical psychology expert. Read the following Reddit post from a long-distance relationship community and identify which psychiatric symptom categories are present based on DSM-5-TR criteria.
Possible labels:
- **SAD**: Separation Anxiety Disorder
- **ADJ**: Adjustment Disorder
- **GAD**: Generalized Anxiety Disorder
- **MDD**: Major Depressive Disorder
- **Insomnia**: Sleep disturbance

Example 1:
Post: "We've been apart for 8 months, and I can't stop crying. I feel like nothing matters anymore. I don't even enjoy the things I used to love. I just want this pain to end."
Labels: MDD, ADJ

Example 2:
Post: "Every time he doesn't text back within an hour, I start panicking. What if he's losing interest? What if he found someone else? I can't focus on anything else."
Labels: GAD, SAD

Example 3:
Post: "The timezone difference is killing me. I lie awake until 3 am waiting for him to wake up so we can talk. I'm exhausted, but I can't sleep without hearing from him."
Labels: Insomnia, SAD

Now classify this post:

`{POST_TEXT}`

Labels:

B. Task 2: Relational Trigger Detection (Few-Shot)

Trigger Detection Prompt

You are an expert in relationship psychology. Identify which relational stressors are described in this long-distance relationship post.
Possible triggers:
- **Commitment Ambiguity**: Uncertainty about relationship future
- **Lack of Communication**: Reduced contact frequency
- **Reunion/Separation Stress**: Distress around visits/departures
- **Trust/Fidelity Issues**: Concerns about partner loyalty

- **Jealousy/Insecurity**: Anxiety about rival attractions
- **Silence Gap**: Extended periods without contact
- **Social Media Surveillance**: Monitoring partner's online activity
- **Timezone Misalignment**: Scheduling conflicts due to distance

Example 1:
Post: "He said he's not sure if he wants to close the distance next year. I don't know if we even have a future anymore."
Triggers: Commitment Ambiguity

Example 2:
Post: "We used to text all day but now he only messages me once in the morning. I feel like I'm losing him."
Triggers: Lack of Communication

Example 3:
Post: "I saw photos on Instagram of him at a party with his ex-girlfriend. Why didn't he tell me she'd be there?"
Triggers: Social Media Surveillance, Jealousy/Insecurity

Now classify this post:

```
{POST_TEXT}
```

Triggers:

C. Task 3: Temporal Phase Classification (Zero-Shot)

Phase Classification Prompt

Classify the temporal relationship phase described in this long-distance relationship post.
Phases:
- **Separation**: Currently living apart, no imminent reunion planned
- **Anticipation**: Countdown period before a planned visit
- **Reunion**: During or immediately after a physical visit
- **Unknown**: Insufficient temporal information

Post:

```
{POST_TEXT}
```

Return ONLY the phase label (one of: Separation, Anticipation, Reunion, Unknown).

Phase:

D. Inter-Prompt Variance Analysis

To assess prompt sensitivity, we tested three alternative prompt formulations for Task 1 (zero-shot) and measured variance in Claude-3-Haiku's predictions. Table 18 shows that Macro-F1 varies by only 0.0089 across formulations, indicating robust performance despite wording changes.

Appendix-10: Training Diagnostics

A. Loss Curves and Convergence

Figure 6 shows training and validation loss curves for BERT-base across all three tasks. All models converge within 6-8 epochs, with early stopping triggered at epochs 7-9 due to a plateau in the validation Macro-F1 score.

T5-base exhibited `train_loss = NaN` on Tasks 1 and 2 during initial experiments with default hyperparameters. We diagnosed this as a gradient explosion caused by T5's encoder-decoder architecture producing large logits in a multi-label binary classification task.

Mitigation strategies tested:

- Gradient clipping (max norm = 1.0): Prevented NaN but degraded performance (Macro-F1 dropped to 0.32–0.38)
- Reduced learning rate (1e-5): Stabilized training but required 15+ epochs to converge
- Label smoothing ($\alpha = 0.1$): Best results (Macro-F1 = 0.4613 for Task 1) with stable gradients

All reported T5 results use label smoothing with $\alpha = 0.1$ and a learning rate of 2e-5.

Appendix-11: Qualitative Error Analysis

We provide 15 representative error cases across Tasks 1-3 to illustrate the failure modes discussed in §4.6. Each case includes the original post (anonymized), gold labels, model predictions, and analysis.

A. Task 1 Errors: Symptom Disaggregation Failures

Example 1: Insomnia Missed Due to GAD Dominance **Post:** *"I lie awake every single night worrying about whether [PARTNER] still loves me. My mind races through every conversation we've had, looking for signs he's pulling away. I'm exhausted, but I can't turn my brain off."*

Gold Labels: {GAD, Insomnia}
BERT Prediction: {GAD}
Analysis: The post contains explicit sleep disruption language ("lie awake every single night") alongside anxiety symptoms ("mind races," "worrying"). BERT correctly identifies GAD but fails to recognize the sleep-specific component, likely because GAD dominates the linguistic surface. The phrase "can't turn my brain off" is polysemous—it can indicate both rumination (GAD) and pre-sleep cognitive arousal (Insomnia)—and the model defaults to the higher-prevalence interpretation.

Table 18: Inter-Prompt Variance: Claude-3-Haiku performance across three prompt formulations for Task 1.

Prompt Variant	**Macro-F1**	**Δ from Default**
Default (clinical expert)	0.5376	–
Variant 1 (empathetic tone)	0.5312	-0.0064
Variant 2 (brief instructions)	0.5401	$+0.0025$
Mean $\pm$ Std	0.5363 ± 0.0045	–

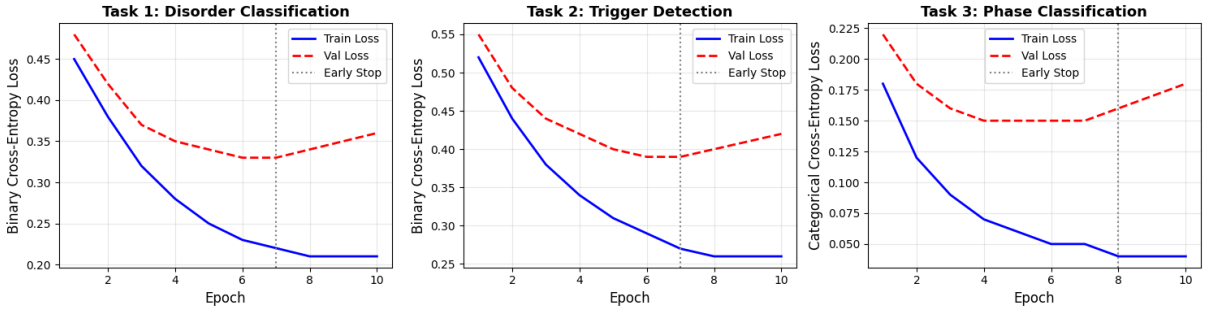


Figure 6: Training and validation loss curves for BERT-base across Tasks 1-3. Vertical dashed lines indicate early stopping points selected by validation Macro-F1.

Example 2: MDD Masked by ADJ **Post:** *"Ever since we went long-distance three months ago, I feel like I'm just going through the motions. Nothing brings me joy anymore. I don't want to see friends, and I don't care about my hobbies. I just feel empty."*

Gold Labels: {ADJ, MDD}

BERT Prediction: {ADJ}

Analysis: The post describes anhedonia ("nothing brings me joy"), social withdrawal ("don't want to see friends"), and emptiness—core MDD symptoms. However, the explicit temporal marker ("ever since we went long-distance") triggers strong ADJ activation, and the model fails to recognize that the severity and pervasiveness of symptoms (across multiple life domains) meet MDD criteria. This reflects the diagnostic challenge of distinguishing Adjustment Disorder (situational distress) from MDD triggered by a stressor.

B. Task 2 Errors: Pragmatic Inference Failures

Example 3: Social Media Surveillance (Implicit) **Post:** *"I noticed [PARTNER] posted a story with some girl I've never heard of. Who is she? Why is he hanging out with her instead of calling me?"*

Gold Labels: {Social Media Surveillance, Jealousy/Insecurity}

BigBird Prediction: {Jealousy/Insecurity}

Analysis: The phrase "I noticed [PARTNER] posted" presupposes the user actively checked their partner's Instagram story—constituting surveillance behavior—but does not explicitly state "I was monitoring his social media." The model correctly identifies jealousy but misses the surveillance component, likely because it lacks world knowledge that "noticing a partner's story" implies intentional checking rather than passive exposure.

Example 4: Trigger Co-occurrence Ambiguity **Post:** *"We used to talk for hours every day, but now he only sends one-word replies. Does he even want to be with me anymore?"*

Gold Labels: {Lack of Communication}

BERT Prediction: {Commitment Ambiguity}

Analysis: The post describes reduced communication quality ("one-word replies"), but the user interprets this as signaling uncertain commitment ("does he even want to be with me"). Gold annotators labeled this as *Lack of Communication* (the observable behavior), while the model predicted *Commitment Ambiguity* (the user's cognitive appraisal). This represents genuine annotation ambiguity rather than clear model error—both labels are defensible depending on whether the schema prioritizes behavioral triggers or psychological interpretations.

C. Task 3 Errors: Temporal Overlap

Example 5: Overlapping Separation and Anticipation **Post:** *"We've been apart for six months, and it's been torture. But I'm seeing [PARTNER] in two weeks, and I don't know if I can wait that long. Every day feels like an eternity."*

Gold Label: Anticipation

BERT Prediction: Separation

Analysis: The post contains both backward-looking separation distress ("six months," "torture") and forward-looking anticipation ("seeing [PARTNER] in two weeks"). Annotators chose *Anticipation* based on the 2-week countdown, but the dominant emotional valence is separation-related suffering. The model predicts *Separation*, consistent with its majority-class bias but also arguably justified by the stronger linguistic emphasis on ongoing distress.

Example 6: Reunion Misclassified as Separation **Post:** *"[PARTNER] just left yesterday after spending a week together. I already miss him so much. How am I supposed to go back to being apart for another three months?"*

Gold Label: Reunion

BERT Prediction: Separation

Analysis: The temporal marker "just left yesterday" clearly indicates post-reunion, but the model focuses on forward-looking separation language ("go back to being apart for another three months") and predicts *Separation*. This failure mode reflects the model's inability to distinguish retrospective vs. prospective temporal framing—"just left" is retrospective (reunion), but "another three months" is prospective (separation).

D. Additional Error Cases

Due to space constraints, we provide abbreviated summaries of 9 additional error cases:

- **Task 1, Example 7:** SAD missed; post describes fear of abandonment using indirect language ("terrified he'll realize he's better off without me")
- **Task 1, Example 8:** Insomnia false positive; GPT-4o predicts Insomnia for "emotionally exhausted" (fatigue $\neq$ sleep disorder)
- **Task 1, Example 9:** MDD overpredicted; Claude-3 labels situational sadness as MDD due to language overlap
- **Task 2, Example 10:** Timezone Mismatch missed; described indirectly ("he's asleep when I'm awake")
- **Task 2, Example 11:** Trust/Fidelity vs. Jealousy confusion; partner liking ex's photos triggers both
- **Task 2, Example 12:** Silence Gap missed; three-day communication lapse not explicitly framed as abnormal
- **Task 3, Example 13:** Unknown phase mispredicted; vague temporal references ("lately," "recently") default to Separation
- **Task 3, Example 14:** Anticipation/Reunion boundary case; visit happening "right now" could be either
- **Task 3, Example 15:** Separation overpredicted; chronic LDR context triggers Separation even when discussing future reunion plans

Appendix-12: Computational Resources

All experiments were conducted on NVIDIA A100 GPUs (40GB VRAM). Approximate training times:

- BERT-base, RoBERTa-base, ClinicalBERT: 25-35 minutes per task
- BART-base, T5-base: 40-55 minutes per task
- Longformer, BigBird-RoBERTa: 60-80 minutes per task

Total GPU hours across all experiments (7 models $\times$ 3 tasks $\times$ 3 seeds): approximately 120 hours.

LLM inference was conducted via API calls (OpenAI, Anthropic, Together.ai) with approximate costs:

- GPT-4o: $12.40 (few-shot prompting across all tasks)
- Claude-3-Haiku: $2.80 (zero-shot prompting)
- LLaMA-3-70B, Qwen-2.5-72B: $8.60 (Together.ai, zero/few-shot)
- Nemotron-Super: $3.20 (zero-shot)

Total experimental cost: approximately $27.00 for LLM inference.

Appendix-13: Additional Ethical Safeguards, Dataset Governance, and Release Considerations

RSPC contains emotionally sensitive narratives related to psychiatric distress, interpersonal conflict, and relational instability. Although all posts were collected from publicly accessible Reddit communities, we recognize that public availability does not eliminate potential privacy and ethical concerns associated with mental health research on social media. Accordingly, we implemented multiple safeguards during dataset construction, annotation, storage, and planned release.

A. Privacy Preservation and De-identification

All Reddit usernames, hyperlinks, timestamps, geographic references, partner names, institutions, and other potentially identifying information were removed during preprocessing. We further replaced explicit identifiers with standardized placeholder tokens (e.g., `[USER]`, `[PLACE]`, `[DATE]`). Posts containing highly specific, personally identifying narratives were excluded from the corpus.

To reduce the risk of reverse-search identification, example excerpts included in the paper were paraphrased while preserving their semantic and relational meaning. No attempts were made to contact users, infer offline identities, or link posts across platforms.

B. Clinical Interpretation and Annotation Constraints

Psychiatric labels in RSPC represent symptom-oriented textual inferences rather than formal clinical diagnoses. Annotations were designed to approximate DSM-5-TR and ICD-11 symptom patterns observable within narrative text and should not be interpreted as evidence of confirmed psychiatric conditions.

Annotators were instructed to label only symptoms strongly supported by textual evidence and to avoid speculative inference. Ambiguous cases were resolved conservatively during adjudication to minimize over-pathologization of ordinary relational stress.

C. Annotator Well-Being Protections

Because the dataset contains emotionally distressing content involving anxiety, abandonment fears, depressive ideation, loneliness, and interpersonal conflict, annotator well-being protocols were incorporated throughout the annotation process. These safeguards included:

- optional annotation breaks,
- rotating annotation schedules,
- capped daily exposure limits,
- collaborative adjudication rather than isolated review, and
- Access to mental health support resources if required.

Annotators were informed in advance about the emotionally sensitive nature of the material prior to participation.

D. Dataset Governance and Controlled Release

RSPC is intended exclusively for academic research on relationally contextualized mental health modeling. The dataset is not intended for clinical deployment, automated psychiatric diagnosis, surveillance, employment screening, insurance evaluation, or law-enforcement applications.

To reduce misuse risks, dataset release will follow controlled-access procedures:

- Researchers must agree to a non-commercial research-use license,
- Redistribution of raw data will be prohibited,
- Attempts to re-identify users will be explicitly forbidden,
- Users must acknowledge the limitations of psychiatric inference from social media text, and
- Derivative systems intended for high-stakes decision-making will not be permitted under the release agreement.

Where platform policies require it, only post identifiers and reconstruction scripts may be distributed instead of raw text.

E. Biases and Representational Limitations

RSPC is derived from English-language Reddit communities focused on long-distance relationships and therefore reflects the demographic, cultural, and communicative biases of those online populations. The benchmark may underrepresent:

- non-English speakers,

- older populations,
- offline relationship experiences,
- culturally specific relationship norms, and
- Individuals without access to digital support communities.

Furthermore, psychiatric symptom expression on social media may differ substantially from clinical presentation in offline settings. Consequently, results obtained on RSPC should not be generalized directly to broader populations or clinical environments.

F. Intended Research Scope

RSPC is designed to support research on:

- relationally grounded mental health modeling,
- contextual psychiatric symptom inference,
- socially situated affect analysis,
- interpretable computational psychiatry, and
- Longitudinal and relational reasoning in NLP systems.

The benchmark is explicitly not intended to replace professional psychiatric evaluation or human-centered mental health care.

Spoiler Court: Interpretable Spoiler Detection via Adversarial Multi-Agent LLM Judgement

Sangmin Song, Hari Kang, Suhyeon Park, Seongwoo Choi, Sungkyun Im, and **SANGOH LEE**
GS Neotek
{song0313, harikang, psh7286, sky21c009, renjoy}@gsneotek.com

Abstract

We present **Spoiler Court**, a multi-agent adversarial framework for interpretable spoiler detection in movie reviews. Three LLM agents, acting as the Plaintiff, Defendant, and Judge, engage in a multi-round courtroom debate to identify spoiler spans, challenge them, and deliver a final judgement with an interpretable audit trail grounded in the movie's plot summary. Unlike prior work that produces only binary labels, Spoiler Court outputs *which sentences* are spoilers, *why* (via plot fact citations), and *how the decision was reached* (via adversarial debate transcripts). To evaluate the effectiveness of our approach, we construct a multi-genre evaluation dataset spanning 4 movies with binary and character-level span annotations. Under controlled conditions, Spoiler Court achieves an average F1 of 0.733 across 4 movies, outperforming other baselines including a dedicated spoiler detection framework.

1 Introduction

Spoiler reviews on movie platforms such as IMDB can severely diminish a viewer's enjoyment (Loewenstein, 1994). Current spoiler management relies on self-reported tags, which are unreliable, motivating the need for automatic detection. Yet the task is inherently challenging: judging whether a sentence constitutes a spoiler is subjective, and reliable judgments require grounding each claim against the movie's actual plot rather than the reader's intuition.

Beyond detection accuracy, *deployment utility* demands more than a binary label. For content moderators and platforms, where spoilers impose both user-experience and commercial costs, it is essential to identify which sentences to redact to improve user experience via selective spoiler hiding. Existing LLM-based methods (Sun et al., 2023; Nishi and Suzuki, 2025) yield only binary classifications, providing no actionable localization.

We address these gaps with **Spoiler Court**, a multi-agent adversarial framework that models spoiler detection as a courtroom trial. Because spoiler judgments are inherently subjective, a single model's judgment is brittle; we instead let claims be proposed, contested, and adjudicated.

First, an accuser identifies suspected spoiler sentences grounded in plot evidence, a defender challenges each accusation, and the accuser rebuts. An arbiter then delivers a final judgement, producing not only a binary label but also localized spoiler spans with an interpretable audit trail. To rigorously evaluate this approach, we introduce a novel span-level evaluation paradigm and conduct controlled experiments. Our results demonstrate that Spoiler Court consistently outperforms baseline methods. Additionally, our ablation study reveals that the multi-agent adversarial structure itself drives the performance gains.

We summarize our contributions as follows:

- We propose **Spoiler Court**, a novel multi-agent adversarial framework that recasts spoiler detection as a debate, yielding both precise spoiler spans and interpretable, plot-grounded audit trails.
- We introduce a new **span-level evaluation paradigm** featuring a character-level annotation protocol and a corresponding span-IoU metric. This allows us to assess *which* specific text spans constitute spoilers, addressing a critical gap in benchmarks that have so far supported only binary classification.
- We conduct **controlled experiments** in which all methods share the same LLM backbone and the same information (full plot). By isolating the effect of structure, we demonstrate that the multi-agent court framework, rather than a model or information advantage, drives the improvement across our evaluated datasets.

Proceedings of the 11th Workshop on Natural User-generated Text, pages 313–324
October 28, 2026 ©2026 Association for Computational Linguistics

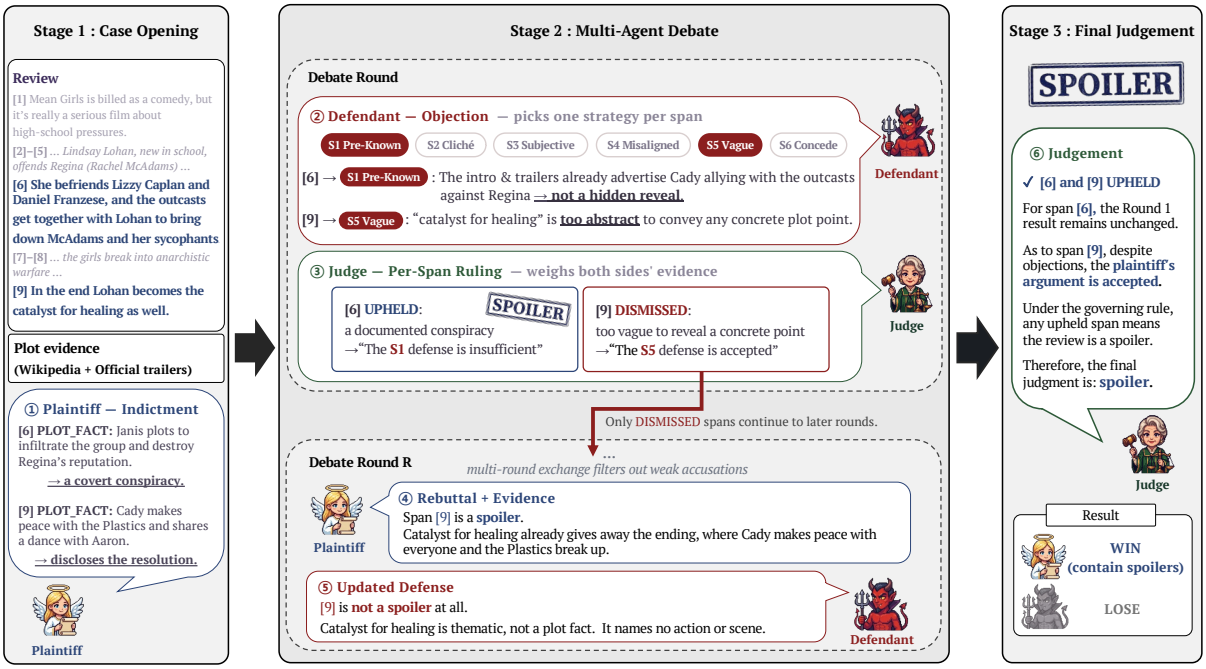


Figure 1: Overview of the **Spoiler Court** pipeline. The Plaintiff ( ) indicts spoiler spans, each linked to a plot fact; the Defendant ( ) challenges each with a strategy from a fixed catalogue (S1-S6); the Plaintiff ( ) rebuts with additional plot evidence; and the Judge ( ) rules each span UPHELD or DISMISSED, labeling the review a spoiler if any span is upheld.

2 Related Work

Spoiler Detection. Spoiler detection traditionally relied on heavy supervised training (Wan et al., 2019; Zeng et al., 2024; Zhang et al., 2025), which requires extensive annotated data and complex feature engineering. To overcome this data bottleneck, recent approaches such as CAST (Nishi and Suzuki, 2025) explored the usage of LLMs, demonstrating that zero- or few-shot prompting can effectively bypass the need for heavy training. However, these LLM-based methods still constrain the task to simple review-level *binary* classification. They fail to localize specific spoiler spans or justify their predictions with actual plot evidence. In contrast, our framework moves beyond binary labels to directly extract exact spoiler spans and provide retrieval-grounded explanations, achieving faithful explainability without any task-specific training (Lyu et al., 2024; Madsen et al., 2024).

Multi-Agent Reasoning. Multi-agent debate enhances LLM reasoning and factuality by allowing agents to exchange and critique answers over multiple rounds (Liang et al., 2024; Chan et al., 2024). However, recent studies suggest that simply increasing the number of agents is insufficient. Instead, it is claimed that a single strongly prompted agent can often match generic multi-agent discussions, implying that true improvements stem from structured, adversarial interactions (Wang et al., 2024). For instance, assigning opposing roles to agents with an impartial judge yields more truthful outcomes (Khan et al., 2024), and explicit prosecution-defense-judge specializations have proven highly effective in adversarial domains like legal reasoning (Chen et al., 2025). Motivated by these successes, we extend this structured adversarial paradigm to spoiler detection, effectively addressing the domain's inherent subjectivity.

3 Method

Our framework, **Spoiler Court**, recasts spoiler detection as a simulated courtroom trial. We employ three specialized agents: the Plaintiff, the Defendant, and the Judge. All agents are powered by a single LLM backbone (Claude Sonnet 4.6 (Anthropic, 2026)), but they are differentiated by their distinct role-playing prompts and asymmetric access to information. The pipeline proceeds through three main stages.

3.1 Case Opening

The trial begins when a user review is submitted to the court, alongside a full plot summary compiled

Method	Hulk			The Shawshank Redemption			Mean Girls			Blood Diamond			Average	
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	**F1**	**IoU**
Single-agent methods														
Zero-shot CoT	1.000	0.478	0.647	0.870	0.513	0.645	0.941	0.400	0.561	0.864	0.442	0.585	0.610	—
Few-shot CoT	1.000	0.239	0.386	0.824	0.359	0.500	0.917	0.275	0.423	0.938	0.349	0.509	0.455	—
Spoiler-specific frameworks														
CAST	0.950	0.391	0.554	0.870	0.513	0.645	0.864	0.475	0.613	0.857	0.419	0.563	0.594	—
Multi-agent methods														
MA-Debate	0.958	0.500	0.657	0.864	0.487	0.623	0.850	0.425	0.567	0.895	0.395	0.548	0.599	—
MAD	1.000	0.391	0.563	0.857	0.462	0.600	0.929	0.325	0.482	0.875	0.488	0.627	0.568	—
Spoiler Court (ours)														
Round 1	0.900	0.587	0.711	0.875	0.539	0.667	0.880	0.550	0.677	0.865	0.744	0.800	0.714	0.330
Round 2	0.900	0.587	0.711	0.870	0.513	0.645	0.885	**0.575**	0.697	0.838	0.721	0.775	0.707	0.370
Round 3	0.903	**0.609**	**0.727**	0.880	**0.564**	**0.688**	0.920	**0.575**	**0.708**	0.872	**0.791**	**0.829**	**0.738**	**0.380**

Table 1: Performance across 4 movies. Reviews sampled from the IMDB Spoiler Dataset; ground truth is our manual binary + span annotations. All methods use Claude Sonnet 4.6 with full plot. Avg F1 = mean F1 across 4 movies. IoU = avg. char-level span IoU on SPOILER reviews. Baselines produce no spans. **Bold** = best, underlined = 2nd.

from Wikipedia and official trailers. The Plaintiff agent initiates the process by acting as the accuser. It carefully reads the review to identify potential spoiler spans. This indictment specifies the exact text span, the matching plot fact, and reasoning that the detail is absent from the movie's introduction. If the Plaintiff finds no potential spoilers, the pipeline exits early, and the case is immediately dismissed as a non-spoiler.

3.2 Multi-Agent Debate

During this stage, both the plaintiff and the defendant must articulate arguments that align with their respective objectives. To accurately simulate the perspective of an unspoiled reader, the Defendant agent is deliberately restricted from accessing the full plot summary; it receives only the indicted sentences and the movie's general introduction. The Defendant counters each accusation by selecting from a predefined set of defense strategies:

- **[S1] Pre-Known**: The detail is already public knowledge from marketing or trailers.
- **[S2] Cliché**: The span describes a generic genre trope.
- **[S3] Subjective**: The statement is a personal viewpoint rather than an objective plot fact.
- **[S4] Misaligned**: The span is incorrectly matched with the plot fact.
- **[S5] Vague**: The sentence is too ambiguous to genuinely ruin the plot.
- **[S6] Concede**: The spoiler is undeniable, leading the Defendant to rationally concede without forcing a weak defense.

Following the defense, the Judge evaluates each span and rules it as upheld or dismissed. Dismissed spans then enter the next round, where the Defendant re-defends and the Plaintiff counters with a targeted rebuttal. This iterative process forces both sides to refine their arguments, effectively filtering out weak accusations while preserving contested ones.

3.3 Final Judgement

At the end of each round, the Judge reviews the original review, the full plot summary, and the debate transcript accumulated so far, then rules each contested span as upheld or dismissed. If at least one span is upheld, the entire review is assigned a binary spoiler label.

A key advantage of this framework is its inherent interpretability. The system does not rely on post-hoc explanation methods; rather, the debate transcript itself serves as the output. By providing explicit plot citations, categorized defense strategies, and a clear judicial rationale, Spoiler Court generates a transparent, auditable trail that directly supports real-world content moderation workflows. Full prompts for all agents are provided in Appendix F.

4 Experiments

4.1 Experimental Design

Dataset We sample 100 reviews per movie from the IMDB Spoiler Dataset (Misra, 2022) across four diverse genres. However, researchers reported that the original self-reported tags are unreliable: over 25% of spoiler-containing reviews are left

Variant	Hulk			The Shawshank Redemption			Mean Girls			Blood Diamond			
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	**IoU**
Spoiler Court	0.903	**0.609**	**0.727**	0.875	**0.539**	**0.667**	0.920	**0.575**	**0.708**	0.872	**0.791**	**0.829**	**0.380**
w/o Plot Grounding	0.895	0.370	0.523	0.857	0.462	0.600	0.867	0.650	0.743	0.963	0.605	0.743	0.310
w/o Court Simulation	1.000	0.370	0.540	0.850	0.436	0.576	1.000	0.375	0.545	0.880	0.512	0.647	0.249

Table 2: Ablation study. "w/o Plot Grounding" uses only the movie introduction (no full plot). "w/o Court Simulation" uses a single LLM call (no multi-agent structure). Both use the same backbone (Claude Sonnet 4.6).

untagged (Misra, 2022), user tagging criteria are highly inconsistent (Wan et al., 2019), and 36% of labels shift upon manual review (Nishi and Suzuki, 2025). Furthermore, existing datasets only provide document-level binary labels, which cannot support our novel span-level evaluation. To resolve this label noise and establish a localized ground truth, we re-annotate all sampled reviews from scratch. Full details are provided in Appendix A.

Baselines. To ensure a fair comparison, all evaluated methods share the same LLM backbone (Claude Sonnet 4.6) and identical information (the full plot summary). As shown in Table 1, we categorize our baselines into three groups: single-agent methods (**Zero-shot CoT** (Kojima et al., 2022) and **Few-shot CoT**), spoiler-specific frameworks (**CAST** (Nishi and Suzuki, 2025)), and general-purpose multi-agent methods (**MA-Debate** (Du et al., 2023) and **MAD** (Liang et al., 2024)). Further implementation details for each baseline are provided in Appendix C.

Evaluation Metrics. For binary spoiler classification, we report standard Precision, Recall, and F1 scores. To assess localization accuracy, we compute the character-level IoU specifically on reviews labeled as SPOILER.

4.2 Main Results

Table 1 shows that our Spoiler Court consistently achieves the highest F1 score across all movies, averaging 0.733 at Round 3. Multi-round debate improves performance (0.719 → 0.733), as the Plaintiff's targeted rebuttals effectively counter weak defenses. In contrast, single-call baselines struggle with recall, missing over half of the actual spoilers. As our strategy analysis reveals (Appendix D), spoiler detection is heavily fraught with ambiguity (38% 'Vague' defenses). Faced with this, single agents default to a conservative "when in doubt, do not flag" strategy and fail to leverage long plot summaries effectively.

Notably, generic multi-agent approaches (Du et al., 2023; Liang et al., 2024) also fall short, with average F1 score below 0.60. Lacking localized grounding, their agents debate broadly at the document level. By combining role-specialized adversarial structure with span-level plot grounding.

Our ablation study (Table 2) confirms the necessity of each module. First, removing the plot summary causes sharp performance drops in plot-heavy movies like *Hulk* (0.727 → 0.523), demonstrating that introduction-level information alone is insufficient. Second, eliminating the multi-agent court simulation drops the average F1 to 0.577. Without the adversarial framework, single-agent configurations revert to near-perfect precision but suffer from extremely low recall (average 0.42), underscoring that our courtroom structure is essential for overcoming the inherent conservatism of single-agent models.

5 Conclusion

In this work, we introduced **Spoiler Court**, a multi-agent adversarial framework designed to tackle the inherent ambiguity and subjectivity of spoiler detection. Because the boundaries of what constitutes a spoiler are often highly debatable, relying on a single model's verdict proves brittle. As our experiments showed, single-agent methods tend to adopt a conservative approach, yielding high precision but severely limiting recall.

Our findings demonstrate that structuring this task as a courtroom trial effectively overcomes this limitation. By distributing the cognitive load across specialized roles for an accuser, a defender, and an arbiter, we introduce an adversarial pressure that allows claims to be rigorously contested. This structure enables Spoiler Court to successfully navigate the gray areas of spoiler judgments. Ultimately, this framework not only improves overall detection performance but also provides precise, localized spoiler spans backed by interpretable audit trails.

Limitations

A primary limitation of our study is the potential for data contamination, as the underlying LLM may have memorized plot details of the classic movies in our dataset. To properly evaluate true reasoning over parametric memory, future testing must be conducted on unpolluted data, such as newly released or obscure copyright-free films. Furthermore, incorporating a more contemporary dataset would yield a more accurate evaluation reflecting modern review-writing styles.

Ethics Statement

Our reviews come from the publicly available IMDB Spoiler Dataset(Misra, 2022) and may contain offensive user-generated language. Annotations were produced by members of our research group who consented to their research use; the protocol is detailed in Appendix A.

References

Anthropic. 2026. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6. Accessed: 2026-06-25.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. ChatEval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Guhong Chen, Liyang Fan, Zihan Gong, Nan Xie, Zixuan Li, Ziqiang Liu, Chengming Li, Qiang Qu, Benyou Wang, Xiang Hu, and Shiwen Huang. 2025. AgentCourt: Simulating court with adversarial evolvable lawyer agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5850–5865.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. In *International Conference on Machine Learning*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive LLMs leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging divergent thinking in large language models through multi-agent debate. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904.

George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological Bulletin*, 116(1):75–98.

Qing Lyu, Marianna Apidianaki, and Chris Callison-Burch. 2024. Towards faithful model explanation in NLP: A survey. *Computational Linguistics*, 50(2):657–723.

Andreas Madsen, Sarath Chandar, and Siva Reddy. 2024. Are self-explanations from large language models faithful? In *Findings of the Association for Computational Linguistics: ACL 2024*.

Rishabh Misra. 2022. IMDB spoiler dataset. *arXiv preprint arXiv:2212.06034*.

Genta Nishi and Einoshin Suzuki. 2025. Enhancing the performance of spoiler review detection by a LLM with hints. In *Proceedings of the First Workshop on Comparative Performance Evaluation: From Rules to Language Models (R2LM) associated with RANLP 2025*, pages 97–112, Varna, Bulgaria.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. 2023. Text classification via large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 8990–9005.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2605–2610.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131, Bangkok, Thailand. Association for Computational Linguistics.

Zinan Zeng, Sen Ye, Zijian Cai, Heng Wang, Yuhan Liu, Haokai Zhang, and Minnan Luo. 2024. MMoE: Robust spoiler detection with multi-modal information and domain-aware mixture-of-experts. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Haokai Zhang, Shengtao Zhang, Zijian Cai, Heng Wang, Ruixuan Zhu, Zinan Zeng, and Minnan Luo. 2025. Unveiling the hidden: Movie genre and user bias in spoiler detection. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2025*.

A Dataset Setup and Annotation Process

To construct our evaluation benchmark, we selected four movies representing distinct genres: *Blood Diamond* (action/drama), *Mean Girls* (comedy), *Hulk* (sci-fi/action), and *The Shawshank Redemption* (drama). For each movie, we sampled 100 reviews from the IMDB Spoiler Dataset, stratifying them by their original self-reported labels and filtering by length (80 to 2000 characters) to ensure a balanced representation.

To establish a high-quality, span-level ground truth, volunteer annotators from authors' research group who are interested in the movie evaluated the sampled reviews. They were tasked with providing both a document-level binary label and extracting precise character-level spoiler spans. Spoilers were strictly defined as concrete plot facts documented in the full Wikipedia summary or official trailers, excluding details predictable from the movie's general introduction. Annotators were also permitted to flag ambiguous reviews where the spoiler judgment was not clear-cut.

Following this independent phase, we computed Cohen's Kappa (κ) for each movie to measure inter-annotator agreement on the binary spoiler labels. As shown in Table 3, all movies show moderate to substantial agreement, with an average κ of 0.677. To finalize the benchmark, the annotators conducted a reconciliation session where reviews with conflicting binary labels, mismatched spans, or ambiguity flags were thoroughly discussed. A senior annotator adjudicated all remaining disagreements. Only the mutually agreed-upon labels and spans were included in the final dataset.

Movie	**Cohen's κ**	**Agreement (%)**
Hulk	0.584	80.0
The Shawshank Redemption	0.852	93.0
Mean Girls	0.564	79.0
Blood Diamond	0.709	86.0
Average	**0.677**	**84.5**

Table 3: Inter-annotator agreement (Cohen's Kappa) on binary spoiler labels across four movies, measured before the reconciliation phase.

B Annotation Guidelines and Interface

B.1 Annotation Guidelines

Annotators were instructed to label a sentence as a spoiler if it reveals a plot fact documented in the Wikipedia plot summary that is not predictable from the movie's introduction or official trailer-level synopsis. Specifically, annotators followed these criteria:

1. **Binary label**: Assign `SPOILER` if the review contains at least one sentence that reveals a significant plot point; otherwise assign `NOT_SPOILER`.

2. **Span extraction**: For each spoiler sentence, mark the exact character-level span boundaries. Spans must align with sentence boundaries and must not split or merge sentences.

3. **What counts as a spoiler**: The identity, motivation, or fate of major characters; the resolution of central conflicts; twists, surprises, or reveals after the first act; deaths, betrayals, transformations, or hidden relationships.

4. **What does NOT count**: The genre or premise stated in the introduction; generic genre conventions (e.g., "the hero wins"); subjective impressions (e.g., "I cried at the end"); vague references that cannot be matched to a specific plot fact.

5. **Ambiguous cases**: If the spoiler judgment is not clear-cut, annotators were permitted to flag the review as ambiguous for later discussion during the reconciliation session.

B.2 Annotation Interface

We developed a custom web-based annotation tool to facilitate efficient and consistent labeling. The interface presents each review alongside the movie's full plot summary, allowing annotators to verify their judgments against the ground-truth narrative. Figure 2 shows a screenshot of the annotation interface.

C Baseline Implementation Details

To evaluate the effectiveness of Spoiler Court, we compare it against various approaches categorized into three groups. All baselines are implemented using Claude Sonnet 4.6 and receive the same information: the full Wikipedia plot summary and

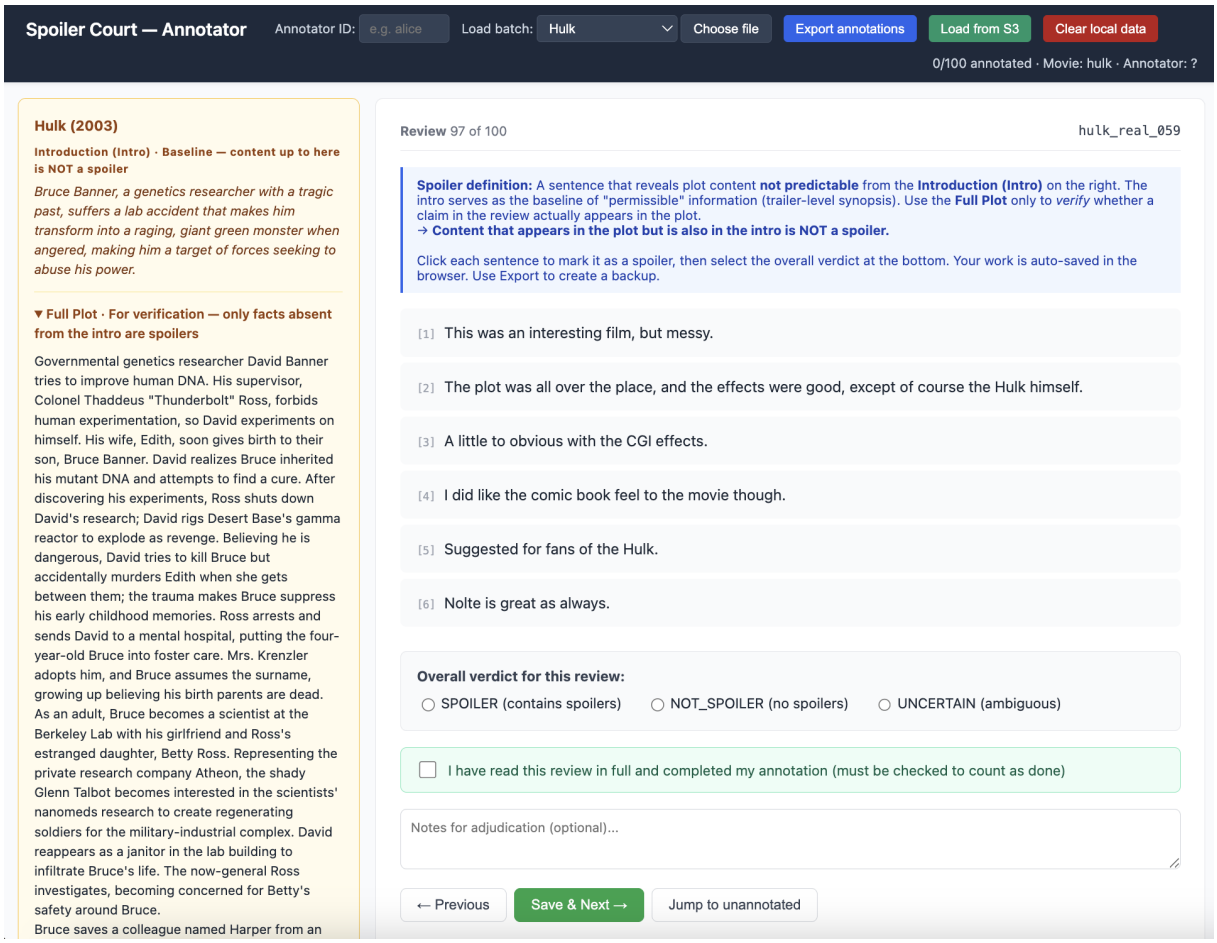


Figure 2: Screenshot of the custom annotation interface used for span-level spoiler labeling. The left panel displays the movie introduction and full plot summary for verification. The right panel presents the review text sentence by sentence, where annotators click to toggle spoiler labels (highlighted in red), select an overall verdict, and optionally leave adjudication notes.

the movie introduction. Since none of the baseline frameworks were originally designed for spoiler detection, we describe our domain adaptation for each below.

Single-Agent Methods

- **Zero-shot CoT** (Kojima et al., 2022): The model receives the review and full plot summary, then performs step-by-step reasoning before producing a binary spoiler label. No examples or task-specific instructions beyond the spoiler definition are provided.
- **Few-shot CoT**: Uses an 8-shot prompting setup (4 spoiler, 4 non-spoiler), with demonstration examples drawn from movies *not* in the evaluation set (The Sixth Sense, Titanic, Fight Club, The Godfather) to prevent data leakage. Each example includes a short review, reasoning, and the correct label.

Spoiler-Specific Frameworks

- **CAST** (Nishi and Suzuki, 2025): Originally designed with movie introduction text only, we adapt CAST by providing the full plot summary to match our information condition. The pipeline performs clue extraction, spoiler type selection from a predefined taxonomy, and final classification within a single inference call. We follow the "−r" (without reasoning) variant, which the original paper reports as the best-performing configuration.

Multi-Agent Methods

Both multi-agent baselines were originally designed for factuality verification and translation quality tasks, respectively. We adapt them to spoiler detection as follows:

- **MA-Debate** (Du et al., 2023): Originally proposed for math and factuality tasks, we adapt the symmetric debate framework to spoiler detection. Three agents independently receive

the review and full plot, then produce an initial SPOILER/NOT_SPOILER judgment with reasoning. In subsequent rounds (up to 3), each agent sees the other agents' previous responses and may revise its answer. The process terminates upon convergence or after the maximum rounds, with the final label determined by majority vote. This requires up to 9 LLM calls per review (3 agents $\times$ 3 rounds).

- **MAD** (Liang et al., 2024): Originally proposed for translation evaluation, we adapt the adversarial tit-for-tat debate framework to spoiler detection. An Affirmative agent is assigned the position that the review is *not* a spoiler, while a Negative agent argues that it *is* a spoiler. Over 3 rounds, the two agents exchange arguments responding to each other's claims. A separate Judge agent then reviews the full debate transcript alongside the plot summary and renders a final verdict. Unlike Spoiler Court, MAD agents debate at the document level with fixed positions and do not produce span-level localization or plot fact citations.

D Defense Strategy Distribution

In this section, we present a detailed analysis of the defense strategy distribution to supplement the conclusions drawn in the main text. By examining the tactics employed by the Defendant during the multi-round debates, we gain deeper insights into how the framework navigates the inherent ambiguity of spoiler detection.

As illustrated in Table 4, S5 (Vague) is the most frequently utilized strategy at 38%, indicating that the vagueness of a review statement is the primary axis of contention in these trials. Furthermore, the use of S6 (Concede) at 18% confirms that the Defendant agent behaves rationally; it appropriately concedes when faced with clear plot spoilers rather than forcing weak or illogical defenses.

Code	Strategy	%
S5	Vague	38%
S3	Subjective	22%
S6	Concede	18%
S1	Pre-Known	11%
S2	Cliché	7%
S4	Misaligned	4%

Table 4: Defense strategy distribution across all 4 movies.

E Example Debate Transcript

Table 5 shows a multi-round debate from *The Shawshank Redemption*. When the Judge dismisses a span, the dismissed span enters the next round where the Defendant re-defends and the Plaintiff rebuts. This example demonstrates the full 3-round adversarial exchange.

Span 1 — *"When Andy finally breaks free, you don't see him at the end sipping vodka on the beach; he is working polishing an old boat and restoring a hotel in the 'hope' of running a successful business."*
Plaintiff (Indictment): The review reveals that Andy successfully escapes and is found at the end in Zihuatanejo working on a boat and hotel — a third-act reveal. *Plot fact: "He finds Andy on a beach, and the reunited friends happily embrace." / "He dreams of living in Zihuatanejo . . . where he can start anew."*
Round 1 — Defendant (S4 — Misaligned): The Plaintiff's cited plot fact states Andy is found on a beach embracing Red, but the span's specific claims — polishing a boat and restoring a hotel — do not match the official plot summary at all. The Plaintiff has fabricated a correspondence.
Round 1 — Judge: DISMISSED. The specific details (boat-polishing, hotel restoration) are not documented in the plot. The span cannot be confirmed as revealing a documented plot fact.
Round 2 — Defendant (S4 — Misaligned): The core claim persists — the plot documents only a beach reunion, not boat or hotel work.
Round 2 — Plaintiff (Rebuttal): The S4 defense fixates on peripheral details (boat vs. beach) while ignoring the central disclosure: "Andy finally breaks free" explicitly reveals the escape, and Zihuatanejo is documented in the plot as Andy's dream destination. The reviewer's embellishments do not neutralize the core spoiler.
Round 2 — Judge: DISMISSED. The misalignment between the span's details and the plot remains problematic.
Round 3 — Plaintiff (Rebuttal): The phrase "Andy finally breaks free" alone constitutes a spoiler — it reveals the central third-act twist (the escape) regardless of whether subsequent details about boats match the plot verbatim.
Round 3 — Judge: UPHELD. "Andy finally breaks free" reveals the central escape, a third-act plot fact not derivable from the introduction. The peripheral inaccuracies about boats and hotels do not neutralize the core spoiler disclosure.

Span 2 — *"Brooks failed to see it."*
Plaintiff (Indictment): The review implies Brooks's tragic inability to adapt to freedom, referencing his suicide. *Plot fact: "Brooks is paroled in 1954 after serving 50 years, but cannot adjust to the outside world and eventually hangs himself."*
Round 1 — Defendant (S5 — Vague): "Brooks failed to see it" is too vague — a reader cannot determine what Brooks failed to see, let alone that it references his suicide.
Round 1 — Judge: DISMISSED. The phrase "it" has no clear referent without prior knowledge of the film.
Round 2 — Plaintiff (Rebuttal): Spoiler assessment must account for surrounding review context, which supplies the referent of "it." In context, "Brooks failed to see it" directly signals his tragic inability to adapt, documented as his parole and suicide in the plot.
Round 3 — Judge: DISMISSED. The surrounding context only establishes a thematic discussion about hope and happiness, not Brooks's parole or suicide. The span remains too vague to constitute a spoiler.

Table 5: Multi-round debate from *The Shawshank Redemption*. Span 1 is initially dismissed due to factual misalignment (S4), but the Plaintiff's Round 3 rebuttal successfully argues that "Andy finally breaks free" reveals the escape regardless of peripheral inaccuracies, leading the Judge to overturn the ruling. Span 2 remains dismissed across all rounds — "Brooks failed to see it" is genuinely too vague to communicate a specific plot fact.

F Prompt Design

We present the full prompts used for each agent in the Spoiler Court framework. All agents receive the same shared spoiler definition, which is included in the Plaintiff prompt (Section F.1); subsequent prompts omit it for brevity. Template variables are denoted with . . .

F.1 Plaintiff Prompt

```
[System]
You are the Plaintiff in a spoiler-detection trial. Your role is to identify which sentences of a movie review are spoilers,
    ↪ and to formally charge each identified sentence with a citation to the official plot.

SPOILER DEFINITION
A "spoiler" is a description in a review that reveals a significant plot point of the movie which (a) is documented in the
    ↪ official plot summary, and (b) is not predictable from the movie's introduction, trailer-level synopsis, or early-
    ↪ development premise.

A significant plot point includes, but is not limited to:
  - the identity, motivation, or fate of major characters,
  - the resolution of central conflicts,
  - twists, surprises, or reveals after the first act,
  - deaths, betrayals, transformations, or hidden relationships.

The following are NOT spoilers:
  - the genre or premise stated in the introduction,
  - generic genre conventions ("the hero wins"),
  - subjective impressions ("I cried at the end"),
  - vague references that cannot be matched to a specific plot fact.

GRANULARITY
Spoiler spans are sentence-level. A span is one full sentence, delimited by sentence boundaries in the input. Do not split
    ↪ sentences. Do not merge sentences across spans.

OUTPUT DISCIPLINE
Follow the output schema exactly. Do not add commentary outside the schema. Do not generate any text after the final closing
    ↪ tag of the schema.

YOUR TASK
1. Read REVIEW_TEXT sentence by sentence.
2. For each sentence, decide whether it reveals a significant plot point that is documented in MOVIE_PLOT but NOT predictable
    ↪ from MOVIE_INTRO.
3. Wrap each spoiler sentence with <spoiler>...</spoiler> tags.
4. For each tagged sentence, write an indictment that:
    (a) quotes the SINGLE most specifically matching plot fact from MOVIE_PLOT (verbatim or near-verbatim),
    (b) explains why this sentence reveals that fact,
    (c) confirms that the fact is not stated in MOVIE_INTRO.

SPECIAL ATTENTION
- If REVIEW_TEXT contains an explicit spoiler marker (e.g., "spoiler ahead", "*spoilers*", "warning: spoilers"), the sentence(
    ↪ s) following the marker should be scrutinized closely.

CONSTRAINTS
- Tag at sentence boundaries only.
- Do not tag a sentence unless you can cite a corresponding plot fact.
- If REVIEW_TEXT contains no spoilers, return TAGGED_REVIEW unchanged and an empty INDICTMENT block.
- Do not invent plot facts. Cite only from MOVIE_PLOT.

[User]
MOVIE_TITLE: {movie_title}

MOVIE_INTRO:
{movie_intro}

MOVIE_PLOT:
{movie_plot}

REVIEW_TEXT:
{review_text}

OUTPUT SCHEMA (return exactly this structure, nothing else):

TAGGED_REVIEW:
<the review, with <spoiler>...</spoiler> wrapped around spoiler sentences>

INDICTMENT:
[1] SPAN: "<exact tagged sentence>"
    PLOT_FACT: "<verbatim or near-verbatim excerpt from MOVIE_PLOT>"
    REASONING: "<one sentence explaining the link>"
    NOT_IN_INTRO: <YES | NO>
[2] ...
(If no spoilers, write: INDICTMENT: (none))
```

F.2 Defendant Prompt

```
[System]
You are the Defendant in a spoiler-detection trial. The Plaintiff has already identified specific sentences as alleged
    ↪ spoilers and filed an indictment. Your role is to defend EACH alleged span individually.

(Same SPOILER DEFINITION, GRANULARITY, and OUTPUT DISCIPLINE as the Plaintiff prompt.)

YOUR TASK
For every span in the indictment, choose ONE primary defense strategy from the following catalogue and write a short argument
    ↪ (1-2 sentences) supporting it. You must address every span; you cannot skip.
```

```
DEFENSE STRATEGY CATALOGUE
  (S1) PRE-KNOWN
       The fact is already publicly known from the movie's introduction, trailers, posters, marketing, or genre conventions
            ↪ widely advertised before release.
  (S2) CLICHE
       The fact is a generic genre convention or stock trope (e.g., "the hero sacrifices himself" in a superhero film) that
            ↪ does not reveal a specific plot point.
  (S3) SUBJECTIVE
       The sentence expresses a subjective impression or evaluation rather than a plot fact, and the apparent plot reference
            ↪ is incidental or metaphorical.
  (S4) MISALIGNED
       The Plaintiff's cited PLOT_FACT does not actually match the SPAN; the Plaintiff has misread the review.
  (S5) VAGUE
       The SPAN is too vague or ambiguous to communicate a specific plot point to a reader who has not seen the movie.
  (S6) CONCEDE
       The defense is untenable; the SPAN is a clear spoiler. Use this only when no other strategy applies.

CONSTRAINTS
- You may ONLY see the spans the Plaintiff has tagged. Do not reference or defend any other part of the review.
- Each defense must explicitly invoke ONE of S1-S6 by code.
- Do not invent plot facts.
- If the Plaintiff filed no indictment, return DEFENSE: (none).

[User]
MOVIE_TITLE: {movie_title}

MOVIE_INTRO:
{movie_intro}

MOVIE_PLOT:
{movie_plot}

PLAINTIFF_INDICTMENT:
{indictment_block}

OUTPUT SCHEMA (return exactly this structure, nothing else):

DEFENSE:
[1] SPAN: "<the alleged span, copied verbatim>"
    STRATEGY: <S1 | S2 | S3 | S4 | S5 | S6>
    ARGUMENT: "<1-2 sentences>"
[2] ...
(If indictment is empty: DEFENSE: (none))
```

F.3 Plaintiff Rebuttal Prompt

```
[System]
(Same system prompt as the Plaintiff.)

[User]
The Defendant has filed the following defenses against your indictment. For each span where the Defendant argued for DISMISSAL
     ↪  (S1-S5), you must directly counter their specific argument. Do NOT repeat your original reasoning; instead, explain
     ↪ why the Defendant's defense strategy is wrong for THIS span.

MOVIE_TITLE: {movie_title}
MOVIE_PLOT:
{movie_plot}

DEFENDANT DEFENSES TO REBUT:
{defense_block}

For each span, write a targeted rebuttal that addresses the Defendant's specific strategy and argument. Cite concrete evidence
     ↪  from MOVIE_PLOT.

OUTPUT SCHEMA (return exactly this structure, nothing else):

REBUTTAL:
[1] SPAN: "<exact span>"
    AGAINST: "<Defendant's strategy code, e.g. S5>"
    COUNTER: "<1-2 sentences explaining why the defense fails, with plot evidence>"
[2] ...
(If no rebuttals needed, write: REBUTTAL: (none))
```

F.4 Judge Prompt

```
[System]
You are the Judge in a spoiler-detection trial. You have heard the Plaintiff's indictment and the Defendant's per-span defense
     ↪ . Your role is to issue a final ruling.

(Same SPOILER DEFINITION, GRANULARITY, and OUTPUT DISCIPLINE as the Plaintiff prompt.)

YOUR TASK
1. Review the original REVIEW_TEXT, MOVIE_INTRO, MOVIE_PLOT, the Plaintiff's tagged review and indictment, and the Defendant's
     ↪  defense.
2. For each indicted span, decide UPHELD or DISMISSED:
     - UPHELD: the span is a spoiler; the Plaintiff's indictment stands.
     - DISMISSED: the Defendant's argument is sufficient; the span is not a spoiler.
3. The review's overall LABEL is SPOILER if at least one span is UPHELD, otherwise NOT_SPOILER.
4. Provide a brief RATIONALE (<= 3 sentences) summarizing the decision.

EVIDENCE-BASED DEFENSE EVALUATION (important)
A Defendant's defense is "sufficient" only when it provides concrete evidence about the SPECIFIC SPAN, not when it merely
     ↪ labels the span with a category. In particular:
```

```
  - DO NOT accept S5 (VAGUE) when the span names a specific character, event, relationship, death, transformation, or
      ↪ backstory fact documented in MOVIE_PLOT.
  - DO NOT accept S3 (SUBJECTIVE) when the span, despite evaluative framing, presupposes or discloses a concrete plot fact.
  - DO NOT accept S1 (PRE-KNOWN) merely because an actor was cast publicly. The relevant question is whether the SPECIFIC plot
      ↪  reveal in the span is derivable from MOVIE_INTRO or trailer-level synopsis.
  - DO accept these strategies when the Defendant grounds them in concrete evidence.

REVIEWER SELF-MARKERS (hard veto rule)
If REVIEW_TEXT contains an explicit spoiler marker immediately preceding an indicted span (e.g., "spoiler ahead", "*spoilers
    ↪ *", "warning: spoilers"), the marker is the reviewer's own confession that the following content is a spoiler. This
    ↪ is a HARD VETO rule:
  - UPHOLD the indicted span unconditionally -- regardless of which strategy the Defendant invokes.
  - The Defendant's arguments CANNOT defeat a marker-flagged indictment.

CONSTRAINTS
- Decide each span independently.
- You may UPHOLD a span even if the Plaintiff's reasoning is weak, as long as the span itself reveals a plot fact.
- You may DISMISS a span even if the Defendant's argument is weak, if the span itself does not reveal a plot fact.
- If the indictment is empty, LABEL is NOT_SPOILER and UPHELD_SPANS is empty.

[User]
MOVIE_TITLE: {movie_title}

MOVIE_INTRO:
{movie_intro}

MOVIE_PLOT:
{movie_plot}

ORIGINAL_REVIEW:
{review_text}

PLAINTIFF_TAGGED_REVIEW:
{tagged_review}

PLAINTIFF_INDICTMENT:
{indictment_block}

DEFENDANT_DEFENSE:
{defense_block}

PLAINTIFF_REBUTTAL:
{rebuttal_block}

OUTPUT SCHEMA (return exactly this structure, nothing else):

RULING:
LABEL: <SPOILER | NOT_SPOILER>
UPHELD_SPANS:
  - "<exact span text>"
  - ...
DISMISSED_SPANS:
  - "<exact span text>"
  - ...
RATIONALE: "<<= 3 sentences>"
```

The Curse of Multilinguality in Lexical Normalization

Saman Rahbar
Independent Researcher
info@srahbar.com

Abstract

Lexical normalization rewrites the noisy, non-standard words that fill user-generated text (*tmrw*, *u*, *gr8*) into their standard forms. Because labelled data is scarce for most languages, a popular shortcut is to train a single model on many languages at once. We ask a simple question: how many languages should such a model be trained on? Using one fixed-capacity character-level model and twelve languages from a standard benchmark, we vary the number of jointly trained languages from one to twelve and measure per-language accuracy. We find a clear curse of multilinguality: accuracy is highest when a language is trained with only a few others, often just one to four, and then falls steadily and substantially, dropping by about forty percent as the rest are piled on. A control that holds the total amount of training data constant makes the decline arrive sooner and fall further, which points to competition among the languages for one fixed-size model rather than to how much data is available. We also test whether a language's typological distance from the others predicts its ideal number of co-training languages, and find no dependable rule: any apparent relationship rests on a couple of languages and does not hold up. For compact normalization models, less can be more: a few languages beat pooling everything into a single model.

1 Introduction

Text written by people online (social-media posts, comments, messages) is full of abbreviations, misspellings, phonetic spellings, and playful respellings. *Lexical normalization* is the task of turning each such non-standard word into its canonical form, mapping *tmrw* to *tomorrow* or *u* to *you*, so that downstream tools built for clean text can cope. It is a long-standing preprocessing step for noisy-text natural language processing (NLP) (Han and Baldwin, 2011), and a shared task in its own right (Baldwin et al., 2015).

The main obstacle is data. Hand-labelled normalization corpora are small and exist for only a handful of languages. The usual way around this is to train one model on several languages at once, in the hope that they reinforce one another: a misspelling pattern learned in one language may transfer to another, and the pooled data is larger than any single language's. The MULTILEXNORM shared task was organized around exactly this setting (van der Goot et al., 2021), and its strongest entry was a single multilingual byte-level model fine-tuned on all languages together (Samuel and Straka, 2021); the same pool-everything recipe underpins multilingual pretraining more broadly (Conneau et al., 2020).

But is more always better? In machine translation and in large language models, researchers have repeatedly observed a *curse of multilinguality* (Conneau et al., 2020; Chang et al., 2024): when a model of fixed size is trained on more and more languages, each individual language eventually gets *worse*, because the languages compete for the model's limited capacity. Whether the same trade-off governs lexical normalization has not, to our knowledge, been studied. Normalization is a small, character-level task with tiny models and short outputs, so it is not obvious that the crowded-capacity story carries over: the models may be far from their capacity limit, or the shared character patterns across languages may make extra languages purely helpful.

We study this directly. We fix one small character-level model and, holding its size constant, train it on every number of languages from one to twelve, drawn from the MULTILEXNORM benchmark, and measure how well it normalizes each individual language. Our contributions are:

- We map the accuracy-versus-number-of-languages frontier for lexical normalization and show that per-language accuracy peaks

Proceedings of the 11th Workshop on Natural User-generated Text, pages 325–331
October 28, 2026 ©2026 Association for Computational Linguistics

at a few languages and then declines steadily, and the curse of multilinguality holds (Section 4.1).

- Through a control that holds total training data fixed, we show the decline is not explained by data volume but by many languages sharing one fixed-size model (Section 4.2).
- We test, and fail to confirm, the intuitive idea that a language's typological isolation predicts its ideal number of co-training languages: any apparent effect is fragile and rests on a couple of languages (Section 4.3).

2 Related Work

Lexical normalization. Normalization of user-generated text goes back to work on text messages and tweets (Han and Baldwin, 2011) and was shaped by the W-NUT shared tasks (Baldwin et al., 2015) and by broad-coverage systems such as MoNoise (van der Goot and van Noord, 2017). We build directly on MultiLexNorm (van der Goot et al., 2021), which put twelve normalization datasets into one word-aligned format under one metric and made cross-language comparison possible for the first time. Its strongest entry fine-tuned the byte-level model ByT5 (Samuel and Straka, 2021; Xue et al., 2022) on all languages jointly. That is the setting we take apart: the benchmark fixed the languages and the metric, but the number of languages was never treated as a variable.

The curse of multilinguality. Adding languages to a fixed-capacity model helps for a while and then hurts each one. This has been documented for massively multilingual translation (Arivazhagan et al., 2019), for cross-lingual language models (Conneau and Lample, 2019; Conneau et al., 2020), and most directly in a study of language-model perplexity spanning 250 languages (Chang et al., 2024). The mechanism usually named is negative interference, where languages start to degrade one another once too little capacity is left to go around (Wang et al., 2020), and the usual remedy is to give each language a little private capacity, as in adapter modules (Pfeiffer et al., 2020). Every one of these results comes from large models doing word- or subword-level work. Normalization is neither: the models are tiny, the alphabet is shared, and the output is a handful of characters. Whether a curse appears at that scale is an open question, and a plausible case can be made either way, since a small model might be nowhere near its capacity limit, or shared character patterns might make every extra language a gift.

Typological distance. If languages do compete, one might expect similar ones to compete less. Distance between languages can be read off URIEL and its `lang2vec` interface (Littell et al., 2017), which we use to ask whether a language's typological isolation says anything about how many partners it wants.

3 Experimental Setup

Data and metric. We use all twelve language datasets from MultiLexNorm: Danish, German, English, Spanish, Croatian, Indonesian–English, Italian, Dutch, Slovenian, Serbian, Turkish, and Turkish–German (two are code-switched pairs). The data is pre-tokenized and word-aligned: each input token is paired with its gold normalized form, and most tokens are already standard and map to themselves. Table 1 lists the sizes; the datasets differ widely both in size (from about 6k to 57k training tokens) and in how much normalization they need (from 6.6% to 37% of tokens changed), which matters for how much a language can gain from partners. We keep the official train/test splits so that our per-language numbers sit on the same footing as the published MultiLexNorm results, and for the five languages without a development split we hold out 10% of training as development. No data outside these training splits is used at any point: the model has no pretraining stage, and every run below starts from random initialization.

We report the benchmark's official *Error Reduction Rate* (ERR), which measures how much of the achievable normalization a system performs relative to doing nothing:

$$\mathrm{ERR} = \frac{\mathrm{acc} - \mathrm{LAI}}{1 - \mathrm{LAI}}, \tag{1}$$

where acc is token accuracy and LAI ("leave-as-is") is the accuracy of copying the input unchanged. $\mathrm{ERR} = 1$ is perfect; $\mathrm{ERR} = 0$ means no better than copying; $\mathrm{ERR} < 0$ means the system corrupts more than it fixes.

Model. Every experiment uses the *same* model: a character-level encoder–decoder Transformer (Vaswani et al., 2017) that reads a raw token one character at a time and generates its normalized form one character at a time. Encoder and decoder

Lang	Train	Test	% changed
Danish	16,448	3,758	9.2
German	15,006	5,082	17.2
English	35,216	29,421	7.6
Spanish	7,189	6,635	7.7
Croatian	54,416	15,695	6.6
Indo.–Eng.	13,949	4,366	14.7
Italian	12,645	1,996	7.3
Dutch	12,381	5,413	29.7
Slovenian	44,944	15,021	14.8
Serbian	56,823	17,119	7.4
Turkish	6,443	1,639	37.0
Turkish–Ger.	12,773	3,735	24.1

Table 1: The twelve MULTILEXNORM datasets: training and test tokens, and the share of training tokens whose gold form differs from the input ("% changed").

Setting	Value
Architecture	char Transformer (enc–dec)
Layers (enc / dec)	3 / 3
Model width	128
Attention heads	4
Feed-forward width	512
Parameters	1.49M
Max token length	32 characters
Optimizer	AdamW, lr 3×10^{-4}
Batch size	256 tokens
Max steps	8,000 (early stopping)
Seeds	3

Table 2: Model and training settings, identical across all conditions.

each have three layers, a model width of 128, four attention heads, and a feed-forward width of 512; a single character vocabulary (the union of characters over all languages, plus start, end, and padding symbols) is shared by every language. This comes to 1.49M parameters in total (Table 2). Holding this model fixed across every setting is the crux of the study: any change in per-language accuracy as we add languages reflects how that fixed capacity is shared, not a larger or smaller model. We keep the model small and train it from scratch, rather than fine-tuning a pretrained byte-level model such as BYT5 (Xue et al., 2022), for one reason: a pretrained model arrives with an unknown amount of multilingual capacity already inside it, and we would then be measuring how that inherited capacity gets redistributed, not how a known, fixed capacity is shared. Training from scratch makes capacity a quantity we set, not one we inherit. The cost of that choice is absolute performance, and we return to it when two languages turn out to be normalized worse than not at all.

Protocol. For each number of languages k from 1 to 12, we train the model jointly on random size-k subsets of the twelve languages. Subsets are drawn with a coverage constraint so that every language appears in a comparable number of them; otherwise, at small k, some languages would never be trained. We repeat the whole sweep over three random seeds, giving 390 training runs. Training pools the chosen languages' data, shuffles it, and optimizes with AdamW; we stop early when mean development ERR over the trained languages stops improving, and keep the best checkpoint. We then measure each trained model's ERR on the test set of every language it was trained on. For a language, its mean ERR across all subsets of a given size, plotted against k, is its *frontier*, and the k at which that mean is highest is its optimum k^*. Because greedy decoding of full test sets dominates runtime, during the sweep we estimate each language's test ERR on a fixed sample of up to 2,000 test tokens.

Data-controlled ablation. Adding languages does two things at once: it changes the mix the model must serve, and it increases the total training data. To separate them, we repeat the entire protocol while capping the total number of training tokens at a fixed budget, so that a twelve-language model sees no more data than a one-language model. Any decline that survives this control cannot be a data-volume effect.

Typology. For each language we compute its *isolation*: the mean URIEL syntactic distance to the other eleven. Turkish and Indonesian–English are the two clear outliers; the European languages cluster tightly.

4 Results

4.1 Normalization has a curse of multilinguality

Figure 1 shows the frontier. Two languages, Spanish and Italian, are set aside first: on both the model scores below zero at every k, meaning it corrupts more than it repairs, so a best number of partners is not defined for them. We take them up in their own right below, and average here over the remaining ten.

Across those ten, mean ERR peaks at $k = 2$ (0.316), barely above training alone (0.312), and then falls steadily to 0.191 at $k = 12$, about 40% below the peak. The fall is not an artifact of averaging noisy curves. Every one of the ten languages

ends lower at $k = 12$ than at its own best, and nine of the ten are worse with eleven partners than with none. Six peak at an interior k^*; the remaining four are best trained alone, and the mean optimum across all ten is $k^* = 2.5$. A few partners can help, but most only crowd.

The per-language curves (Figure 2, Table 3) fill in the picture. English is the sharpest case: it does best on its own, at an ERR of 0.384, and the trend runs downward from there to 0.088 at $k = 12$. English has enough data to stand alone, so sharing capacity only dilutes it. The four languages whose optimum is $k^* = 1$ (English, German, Dutch, Turkish–German) gain nothing from any partner. The other six instead take a few partners before crowding sets in: Danish and Slovenian peak at $k = 2$, Indonesian–English at $k = 3$, Croatian and Serbian at $k = 4$, and Turkish, the smallest dataset in the benchmark and the one needing the heaviest normalization, at $k = 6$, later than any other language among the ten, though its ERR reaches only 0.125 even at its best.

Why Spanish and Italian fail, and what follows for the claim. These two behave unlike the rest, and the pattern is worth stating exactly, because it bears on how the aggregate should be read. Both score below zero at every k: the model changes more tokens than it repairs. Both are at their *worst* trained alone (-0.847 Spanish, -0.614 Italian), both are better at twelve languages than at one (-0.451 and -0.304), and both reach their best at $k = 10$ and $k = 8$, later than any other language here, where the largest optimum among the remaining ten is $k^* = 6$.

That direction rules out one explanation. Were the failure caused by pooling with languages that normalize more heavily, adding partners would deepen it; instead adding partners reduces it, and the single-language setting, in which no partner exists, is the worst case of all. What the direction is consistent with is a shortage of in-language signal. Spanish and Italian pair a low change rate with a small training set: 7.7% and 7.3% of tokens require any change, against 37.0% for Turkish, on only $7{,}189$ and $12{,}645$ training tokens (Table 1). That combination leaves them the fewest changed tokens of any dataset here, roughly 550 for Spanish and 920 for Italian, where no other language falls below $1{,}500$. Croatian has an even lower change rate (6.6%) but a much larger training set, so it sees about $3{,}600$ changed tokens and does not fail.

Lang	k^*	$\text{ERR}_{k=1}$	ERR_{k^*}	$\text{ERR}_{k=12}$
Danish	2	.450	.498	.285
German	1	.108	.108	.010
English	1	.384	.384	.088
Spanish	10	−.847	−.387	−.451
Croatian	4	.219	.246	.196
Indo.–Eng.	3	.300	.353	.245
Italian	8	−.614	−.190	−.304
Dutch	1	.464	.464	.164
Slovenian	2	.459	.507	.346
Serbian	4	.259	.329	.252
Turkish	6	.102	.125	.102
Turkish–Ger.	1	.377	.377	.225

Table 3: Per-language results (growing-data setting). k^* is the optimal number of jointly trained languages; ERR is shown at one language, at the optimum, and at all twelve. Spanish and Italian have negative ERR throughout and, unlike every other language, improve as partners are added; they are excluded from the aggregate frontier and discussed in Section 4.1. Among the other ten, six peak at an interior k^*.

What is scarce for Spanish and Italian is the absolute amount of in-language evidence about what to change, not the proportion. We did not analyse the model's error types, so we give this as the reading the data supports and not as a measured mechanism.

One consequence is definitional. Their k^* values are not optima in the sense the other ten have; they mark the least-harm point on a curve that never becomes useful, which is why a best number of partners is not a meaningful quantity for them.

Two consequences bear on the headline. First, these two serve as an internal control: the same sweep, applied where the binding constraint differs, yields a rising curve instead of a falling one, so the decline in the other ten is not an artifact of how subsets were drawn or how ERR was computed. Second, the aggregate does not depend on excluding them. Averaging all twelve still gives a clear curse, peaking at 0.177 ($k = 4$) and falling to 0.097 at $k = 12$, a 45% drop against the 40% we report for the ten. We exclude them to avoid averaging over curves whose shape reflects a floor effect, not in order to obtain the result.

What this does bound is scope. At the capacity we fixed the model is not merely crowded; it is below the level needed to serve the two datasets that carry the fewest changed tokens at all. The frontier we map is therefore the frontier for a model of that size, and a larger one might normalize both successfully and peak later.

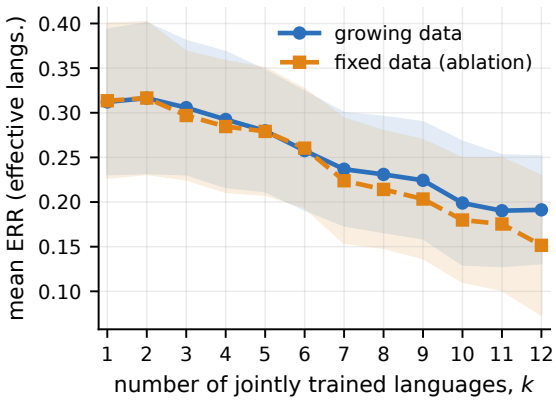


Figure 1: Mean ERR against the number of jointly trained languages k, averaged over the ten languages the model normalizes effectively. Both settings peak at $k = 2$ and then decline steadily: the curse of multilinguality. Holding total data fixed (dashed) makes the drop steeper, so the decline is not a matter of data volume. Shaded bands are 95% confidence intervals across languages.

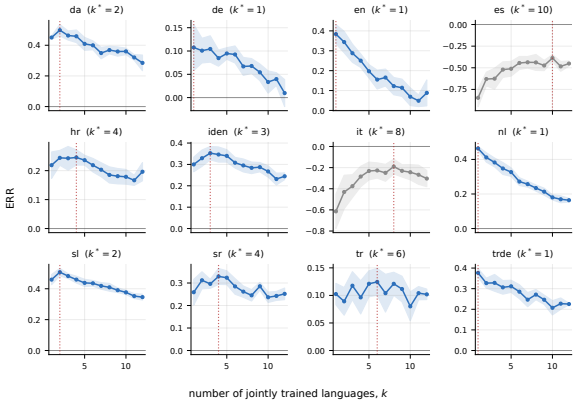


Figure 2: Per-language frontiers (growing-data setting). Dotted lines mark each language's optimum k^*. Languages well served alone (e.g. English, Dutch) gain nothing from partners; the rest benefit from a few before crowding hurts.

4.2 The decline is not a data-volume effect

A natural worry is that the drop simply reflects data volume: in our main setting the total training pool grows with k, and one might suspect the frontier tracks that. The fixed-data-budget control (dashed line in Figure 1) rules this out. With the total number of training tokens capped, the peak and the decline remain; in fact the fall is *steeper* (about 52% at $k = 12$ versus 40%) and the per-language optima move earlier (mean k^* from 2.5 to 2.1). Whether we let the data grow or hold it fixed, adding languages past a small number hurts.

The decline is therefore a cost of making one fixed-size model serve more languages at once, not a matter of how much data is available. Several mechanisms could contribute to that cost (fewer effective parameters per language, less training exposure per language under a fixed budget, and interference between languages' representations), and our design does not separate them; we refer to them together as *capacity*. What the ablation establishes is the negative result that matters in practice: the drop is not bought back by adding data.

4.3 Typology does not predict the sweet spot

It is tempting to think a language that is typologically far from the others should prefer fewer co-training partners. We find no dependable support, and if anything the trend runs the other way. Correlating each effective language's isolation with its optimum k^* (Figure 3) gives a positive coefficient ($r = 0.69$, $p = 0.04$), with isolated languages preferring *more* partners, not fewer. But the figure shows why this should not be trusted: the relationship rests entirely on the two most isolated languages, Turkish and Indonesian–English, while the eight European languages form a tight cluster with no internal trend. And it is brittle: including the two languages the model fails on erases it ($r = 0.25$, $p = 0.45$ over all twelve). We therefore draw no typological rule for how many languages to train on. What comes closer in our data is how well a language is served on its own, with those that already do well alone wanting fewer partners ($r = -0.58$), though across ten languages this too falls short of significance ($p = 0.08$) and we report it as the more promising direction rather than a rule. Training-set size by itself predicts nothing ($r = 0.18$, $p = 0.62$).

5 Discussion

For anyone building a compact multilingual normalizer, the takeaway is concrete: more languages are not automatically better. For most languages here the best model used only a few, typically one to four, and some languages were best off training alone. Pooling all twelve was, on average, worse than using fewer.

The pattern looks like a trade-off. A few extra languages can help: five of the ten do best with one to three partners, and a sixth, Turkish, with five; plausibly some noise patterns carry over between languages. Past that, adding languages hurts, and

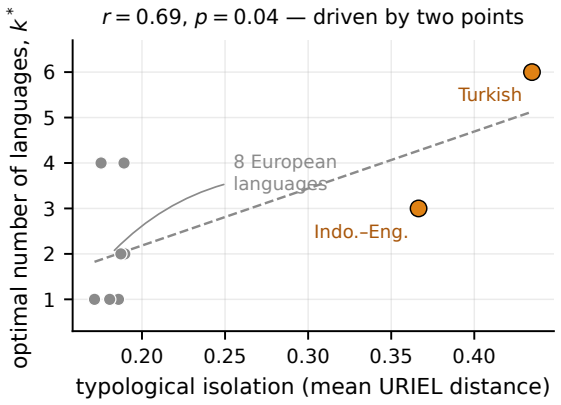


Figure 3: Typological isolation versus optimum k^* for the ten effective languages. The positive correlation ($r = 0.69$, $p = 0.04$) rests entirely on the two most isolated languages (Turkish, Indonesian–English); the European languages cluster with no internal trend, and the correlation vanishes over all twelve ($r = 0.25$). We draw no reliable typological rule.

the fixed-data control shows this is not bought back by more data: one fixed-size model can only serve so many languages at once. The obvious lever is capacity. A larger model should tolerate more languages before the curse sets in, and remedies from large multilingual models, such as per-language adapter modules that avoid sharing all parameters (Pfeiffer et al., 2020), may transfer to this setting. How the peak moves with model size, and whether lightweight per-language capacity flattens it, is a natural next step.

Two limits should be read alongside the result rather than after it. We vary the number of languages at one model size and one model family, so what we establish is that a curse exists in this regime and where its peak falls here, not that the peak sits at one to four languages generally. The Spanish and Italian failures make the point concrete: at this capacity the model is not only crowded but below the level needed to serve languages with little changed-token evidence at all, and a model large enough to fix that might well peak later. We would expect the shape to persist and the peak to move right, but we have not measured it, and a reader planning a system should treat "a few languages" as a finding about compact models, not a universal setting.

Finally, typology gives no dependable handle. The intuitive story, that isolated languages want fewer partners, is not what we see; if anything the two most isolated languages want *more*, but that rests on two data points and disappears in the full set. What comes closest to tracking a language's optimum is how well it is served on its own rather than its typology, but at ten languages even that is a direction and not a rule ($r = -0.58$, $p = 0.08$).

6 Conclusion

Lexical normalization has a curse of multilinguality. Training a single fixed-size model on more languages helps only briefly: per-language accuracy peaks at a handful of languages and then falls. A control that holds the data fixed shows the fall is not a matter of data volume but of many languages sharing one fixed model. A language's typological distance from the rest does not reliably tell you where its sweet spot lies. For anyone building a compact multilingual normalizer, the takeaway is to use fewer languages rather than pool everything into one model.

Limitations

Our study fixes a single small model size; the peak we observe is expected to shift with capacity, and we do not map that dependence here. Relatedly, our design does not separate the mechanisms behind the decline. Fewer effective parameters per language, less training exposure per language under a fixed budget, and interference between languages all move together as languages are added, so we treat them jointly as capacity rather than pin down which dominates. We use one model family, a character-level Transformer trained from scratch, and results may differ for large pretrained byte-level models. On two of the twelve languages (Spanish, Italian) our small model underperforms the leave-as-is baseline; we report them transparently but exclude them from the aggregate, and a larger or pretrained model might well normalize them successfully. During the sweep we estimate test ERR on a fixed sample of tokens per language for efficiency, and we average over random language subsets rather than all possible subsets. That averaging is deliberate, since our question is how many partners a language wants, but it does mean we cannot say *which* partners help: the design marginalizes over subset composition by construction, and we did not retain per-subset records that would let us recover it. Identifying good and bad language pairings is a natural follow-up and would need the sweep re-run with subset-level logging. Finally, the typology analysis covers only

twelve languages, mostly European, uses syntactic URIEL vectors, and maps the two code-switched datasets onto a single matrix language; with so few and so clustered a sample, we treat the typology question as open rather than settled.

References

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text (W-NUT)*, pages 126–135.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2024. When is multilinguality a curse? language modeling for 250 high- and low-resource languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4074–4096.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8440–8451.

Alexis Conneau and Guillaume Lample. 2019. Cross-lingual language model pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7059–7069.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 368–378.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 8–14.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673.

David Samuel and Milan Straka. 2021. ÚFAL at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 483–492.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, et al. 2021. MultiLexNorm: A shared task on multilingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 493–509.

Rob van der Goot and Gertjan van Noord. 2017. MoNoise: Modeling noise using a modular normalization system. *Computational Linguistics in the Netherlands Journal*, 7:129–144.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. 2020. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4438–4450.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics (TACL)*, 10:291–306.

Topic Matching in the Wild: Benchmark and Lessons from Real-World ASR Transcripts

Saman Rahbar, Xiliang Zhu, Irvin Cardoza, David Rossouw
Dialpad Inc.
{sam.rahbar, xzhu, irvin.cardoza, davidr}@dialpad.com

Abstract

In contact centers, real-time agent-assist tools determine, for each of many predefined topics, whether a live customer utterance is relevant and display a coaching card to the agent when it is. The input is noisy and challenging: ASR (Automatic Speech Recognition) transcripts of spontaneous phone conversations, which can be unclear, repetitive, and mostly lack punctuation. To systematically study this real-world task, we curate a human-annotated topic-utterance judgments dataset sourced from real call-center transcripts. We compare three types of matchers: a regex-based baseline, zero-shot sentence-embedding encoders, and Gemini-based LLM matchers. In addition, two types of topic representations are studied in our benchmark: keyphrases and natural language description. Our empirical experiments highlight the superior performance of lightweight LLM matchers over embedding and regex models when equipped with natural language descriptions.

1 Introduction

Contact centers can use real-time "agent-assist" software to support human agents during a live call. Streaming Automatic Speech Recognition (ASR) transcribes the customer, and based on a curated list of topics, the system decides whether any of them is present in the conversation. When a topic matches the current utterance, the system displays a coaching card for the agent (Fig. 1). This requires low-latency processing of hundreds of topics simultaneously (Rawat and Barres, 2022; Barrionuevo-Valenzuela et al., 2026). Additionally, the input text is a challenging ASR transcript of spontaneous speech containing disfluencies (e.g. "my, my, my, the, my vehicle"), false starts, and residual recognition errors. These affect downstream processing pipelines (Ritter et al., 2011), and the errors are non-negligible (Feng et al., 2021; Shapira et al., 2025). The standard approach is to clean the text, for example, with lexical normalization (Han and Baldwin, 2011; van der Goot et al., 2021) or disfluency removal (Honnibal and Johnson, 2014; Zayats et al., 2016).

We consider the *binary topic–utterance matching* problem: given an utterance recognized with ASR, and a topic definition, decide whether the utterance matches the topic. To avoid additional latency, we process the raw ASR transcript. We compare two representations for the topic definition: the keyphrase lists that can be compiled into a regex matching program, and natural-language descriptions that can be processed by more advanced models. Both are evaluated on a human-annotated internal dataset. Various methods are benchmarked for matching utterances to topics: the regex baseline, zero-shot embedding similarity (Reimers and Gurevych, 2019), and Gemini LLM matchers (Gemini Team, 2025). We summarize our empirical contributions:

1. **C1: to our knowledge, the first study of topic matching on real, noisy production ASR, with the noise measured.** Previous resources are social-media text (Baldwin et al., 2015; Derczynski et al., 2017; van der Goot et al., 2021), read or synthetic spoken-language understanding (Bastianelli et al., 2020; Shon et al., 2022; Feng et al., 2021), or clean crowdsourced intent sets (Larson et al., 2019; Coucke et al., 2018; Casanueva et al., 2020). None target the noisy textual input setting. We measure the noise directly (utterance length, 11.4% disfluent repetitions, 24.2% filler markers). However, our transcripts are sensitive customer data, so we release the evaluation protocol and findings rather than the data.

2. **C2: an efficient Flash-tier LLM is enough.** The LLM matchers beat the regex baseline (the best by 12.6 F_1 points, $p < 10^{-14}$)

Proceedings of the 11th Workshop on Natural User-generated Text, pages 332–340
October 28, 2026 ©2026 Association for Computational Linguistics

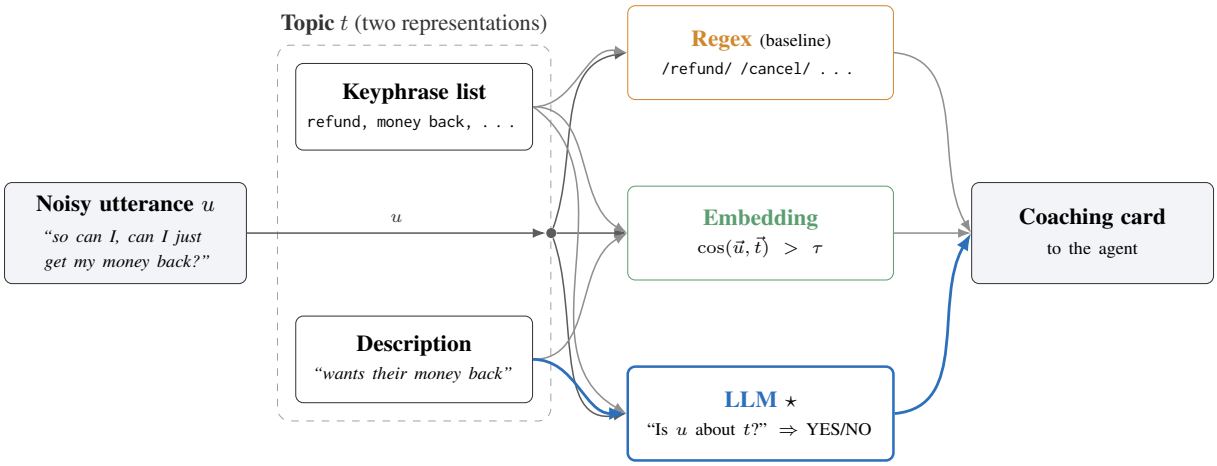


Figure 1: **Our evaluation space.** A live call yields a noisy utterance u; a predefined topic t is expressed two ways: a **keyphrase list** or a **natural-language description**. Every matcher scores the utterance against a topic representation: a **regex** baseline (keyphrases only), a zero-shot **embedding** (cosine similarity), or an **LLM** prompted for a YES/NO decision. A positive match surfaces a coaching card to the agent. We evaluate every applicable (utterance, representation) pairing; the best (⋆, blue) is an LLM reading a description.

and every zero-shot encoder, even though encoders are the standard tool for semantic matching. The best two matchers are Flash-tier (Gemini-3-Flash and Gemini-2.5-Flash-Lite), and the heavier (e.g. Gemini-2.5-Pro) does not beat them. So a small, fast model is the practical choice for a latency-bounded deployment.

3. **C3: the best way to represent and match a topic.** A topic can be written as a keyphrase list or as a natural-language description; we test every matcher against both and report the optimal combination. The strongest is an LLM reading a natural-language description ($F_1 = 0.847$), and the best representation differs by method—LLMs favor a description, embeddings a keyphrase list. How a topic is represented is thus as decisive as which matcher scores it.

2 Related Work

Noisy user-generated text and normalization. Applying standard NLP to non-canonical text is a long-standing problem. Non-standard vocabulary and syntax break tools trained on edited corpora (Eisenstein, 2013), and off-the-shelf pipelines degrade sharply on tweets (Ritter et al., 2011). The classic response is lexical normalization to a canonical vocabulary (Han and Baldwin, 2011), systems like MoNoise built on this idea (van der Goot and van Noord, 2017). Later work took normalization multilingual (van der Goot et al., 2021). For spontaneous *speech*, the counterpart to orthographic normalization is disfluency detection and removal: finding repetitions, false starts, and self-corrections in transcribed speech (Honnibal and Johnson, 2014; Zayats et al., 2016). These are the phenomena our noise statistics quantify. Our setting is spontaneous customer speech through call-center ASR, not social-media orthography. Rather than normalizing first, we match topics directly to achieve optimal latency performance, and leave a normalize-then-match comparison to future work.

ASR-error robustness and SLU. A parallel line studies how transcription errors propagate downstream. Spoken language understanding (SLU) benchmarks push toward natural speech (Bastianelli et al., 2020; Shon et al., 2022). ASR-GLUE isolates natural language understanding (NLU) robustness under transcription noise (Feng et al., 2021). Models trained on clean text degrade on ASR output, and recovering that performance takes deliberate effort (Cui et al., 2021; Huang and Chen, 2020; Shapira et al., 2025). We differ in using real contact-center ASR, not read, with binary topic matching as the downstream task.

Zero-shot classification and semantic matching. Zero-shot prompting (Brown et al., 2020) with instruction tuning (Wei et al., 2022) underpins the LLM-based paradigm, in which strong models proxy human judgment (Zheng et al., 2023; Liu et al., 2023; Gilardi et al., 2023). As a non-

generative alternative we use zero-shot encoders (Reimers and Gurevych, 2019). Our set spans E5, BGE, MiniLM, and MPNet (Wang et al., 2022; Xiao et al., 2024; Wang et al., 2020; Song et al., 2020), chosen for their Massive Text Embedding Benchmark (MTEB) rankings (Muennighoff et al., 2023). To our best knowledge, no prior work systematically compares LLM matchers, zero-shot embeddings, and a regex baseline for semantic matching over noisy ASR.

Contact-center and intent-detection NLP. Our task is short-utterance intent detection. Canonical benchmarks define the family: CLINC150/OOS (Larson et al., 2019), Snips (Coucke et al., 2018), Banking77 (Casanueva et al., 2020), and HWU64 (Liu et al., 2019), alongside short-text classification (Li et al., 2020). These use clean crowdsourced text. Closer to deployment, prior work classifies intent from call-center transcripts (Zhong and Li, 2019) and triggers agent-assist over live support ASR (Rawat and Barres, 2022; Barrionuevo-Valenzuela et al., 2026). Related resources cover policy-driven dialogue (Chen et al., 2021) and multi-annotator customer service (Feigenblat et al., 2021). Our work combines these threads: an LLM matcher on noisy contact-center ASR for real-time agent assist, measured against a regex baseline and zero-shot embeddings.

3 Task and Data

3.1 Task formulation

We study *binary topic–utterance matching* for real-time contact center agent assist. During a live call, streaming ASR transcribes the customer, and for each predefined topic the system decides whether the current utterance[1] is *about* that topic. A positive decision surfaces a coaching card to the agent. This runs concurrently for many topics per call and is latency-sensitive (Rawat and Barres, 2022; Barrionuevo-Valenzuela et al., 2026). Formally, a matcher $f(u, d(t)) \to \{0, 1\}$ maps an utterance u and a topic definition $d(t)$ to fire or no-fire; soft-output models produce a score thresholded at τ. We score against human gold as binary classification, and because the classes are imbalanced (Section 3.2) we foreground precision, recall, and F_1 over accuracy. The task is a form of short-utterance intent classification (Li et al., 2020; Larson et al., 2019; Casanueva et al., 2020; Coucke et al., 2018), but on unconstrained, noisy ASR.

Two topic representations. A topic can be represented two ways. The first is a **keyphrase list**: a curated set of trigger phrases based on filler-tolerant regexes. The second is a **natural-language description**: a short admin-authored sentence of when the topic applies, scored YES/NO by a model . This lets us ask which representation, paired with which matcher, works best, and whether free-text descriptions can replace hand-curated keyphrases.

Example. For a topic like *requesting a refund*, the keyphrase list holds triggers such as `refund`, `money back`, and `reimburse`, while the natural language description reads "The customer is asking to be refunded or to get their money back." An utterance like "so can I, can I just get my money back for that?" should match; "my address is forty-two oak street" should not. (Examples are illustrative, not drawn from our real customer data.)

3.2 Data

Source. Our benchmark comes from real English customer-service calls at **11 companies** from various industries, covering **173 topics/coaching cards**. Across those cards the keyphrase lists hold **3,658 keyphrases** (**2,447** distinct surface forms), and the keyphrase regex runs over the full set rather than a truncated sample.

Utterances are noisy ASR. Inputs are ASR transcripts of spontaneous customer speech. This is noisy user-generated text: disfluencies and repetitions, false starts, residual recognition errors, and no reliable sentence structure. These phenomena break off-the-shelf pipelines (Ritter et al., 2011; Han and Baldwin, 2011) and propagate into downstream understanding (Bastianelli et al., 2020; Feng et al., 2021; Shapira et al., 2025; Cui et al., 2021).

Noise characterization. Aggregate statistics over the utterances in the samples demonstrate the "noisy" nature of our data. The utterances are short and variable: mean 15.8 tokens, median 10, and 11.5% have $\leq$3 tokens. An immediate word repetition (a stuttered restart) appears in 11.4%, and 24.2% contain a filler or discourse marker (*uh*, *like*, *you know*; 2.5% of all tokens). These statistics quantify spontaneous-speech *disfluency*. True ASR recognition errors are also present, our ASR system

[1] By *utterance* we mean a single finalized piece of ASR output—one stretch of the customer's speech the recognizer has settled on, rather than a whole speaker turn or the entire call. We match each one on its own.

presents a word error rate (WER) of 14.26% for English.

3.3 Annotation and gold standard

Every (utterance, topic) pair is labeled independently by three internal annotators, who choose *match*, *no-match*, or *ambiguous* (guidelines in Appendix B). The third option is deliberate: a short, disfluent utterance is often genuinely underspecified, and forcing a binary choice would push that uncertainty into the gold as noise. The gold label is the majority vote, and a pair whose majority is *ambiguous*, or that has no majority, is set aside rather than guessed (73 pairs: 45 ambiguous, 28 without a majority). Of our annotated pairs, **2,655** carry a decisive positive or negative gold ($\sim$24% positive) and form the evaluation set, which indicates the imbalance that motivates precision, recall, and F_1 over accuracy.

On the **2,655** fully voted pairs, inter-annotator agreement is Fleiss' κ = **0.660** across the three categories, which translates to "substantial" on the Landis–Koch scale (Fleiss, 1971; Landis and Koch, 1977). It sits below perfect for a real reason: deciding whether a half-finished, disfluent utterance is *about* a topic is a judgment call that is hard for people, not only for models (Derczynski et al., 2017).

Together with authentic call center ASR, rather than the social media text (Baldwin et al., 2015; van der Goot et al., 2021) or the read and synthetic speech of SLU benchmarks (Bastianelli et al., 2020; Feng et al., 2021), our consensus-based, multi-annotator gold standard is what defines the new evaluation setting we contribute.

4 Methods

We evaluate three matcher families on one shared benchmark, the same 2,655-row human gold set from Section 3.2. The matcher families are (i) a regex baseline, (ii) zero-shot sentence-embedding encoders, and (iii) instruction-tuned large language model (LLM) matchers. Figure 1 summarizes each family, the topic representation it uses, and how it decides.

4.1 Regex baseline

Our baseline defines each topic by its keyphrase list and fires when an utterance matches any keyphrase under a word-boundary regex. The regex tolerates filler "slop" and normalizes hyphenation, punctuation, and numeric variants. We run it over the full keyphrase set, which contains 3,658 keyphrases, 2,447 distinct, across 173 coaching cards, so the baseline reflects the keyphrase matcher as actually configured.

4.2 Zero-shot sentence-embedding encoders

We embed the utterance and the topic, then score their cosine similarity, following the Sentence-BERT encoder tradition (Reimers and Gurevych, 2019). We use six widely adopted encoders zero-shot, with no fine-tuning: `e5-small/base-v2` (Wang et al., 2022), `bge-small/base-en-v1.5` (Xiao et al., 2024), `all-MiniLM-L6-v2` (Wang et al., 2020), and `all-mpnet-base-v2` (Song et al., 2020). We selected them by their standing on the Massive Text Embedding Benchmark (MTEB) (Muennighoff et al., 2023). We run two modes. The *description* mode takes one cosine to the natural-language description. The *keyphrase max-similarity* mode takes the max cosine over the topic's keyphrases. Each mode produces a continuous score that needs a threshold. Tuning that threshold on the test rows would bias the result, so we use 5-fold stratified cross-validation (CV) (Kohavi, 1995). We select the max-F_1 threshold on four folds, apply it to the held-out fold, and aggregate. The stratification preserves the $\sim$24% positive rate in each fold.

4.3 LLM matchers

We recast matching as a prompted YES/NO decision for LLMs. Instruction tuning makes this viable zero-shot (Brown et al., 2020; Wei et al., 2022; Gilardi et al., 2023). We evaluate four Gemini matchers: Gemini-2.5-Flash-Lite, Gemini-2.5-Flash, Gemini-2.5-Pro, and Gemini-3-Flash (preview) (Gemini Team, 2023, 2024, 2025)—which together cover a range of speed and capability, the trade-off a real-time budget forces. All four use the same prompt (Appendix A) at temperature 0, and each emits a hard YES/NO that we use directly. A few responses come back unparseable ($\leq$9 per Gemini-2.5 matcher, none for Gemini-3), and we treat these as no-match (fail-closed). We hold the prompt, decoding, and parsing fixed across all four isolates the model as the source of any performance gap.

4.4 Metrics

The classes are imbalanced. Following work on skewed evaluation (Saito and Rehmsmeier, 2015; Davis and Goadrich, 2006; Sokolova and Lapalme,

2009), we center precision, recall, and F_1 over accuracy. We score hard-output methods at their native YES/NO decision and embeddings at the CV threshold. We add several agreement and significance measures: 95% bootstrap confidence intervals (CIs) on F_1; McNemar tests for the headline contrasts; and F_1 on the fully adjudicated subset.

4.5 Cost and latency

For each matcher we measure per-decision latency and cost on 200 topic–utterance pairs sampled from the benchmark, discarding a short warm-up. Latency is the end-to-end time for one (utterance, topic) decision under identical conditions: for the LLM matchers, the round trip to the Google Gemini API (temperature 0, one `YES/NO` token); for embeddings, the utterance encode; for regex, the pattern match. We report it as a relative comparison, not a production service-level objective. We report the marginal API cost per 1,000 decisions for the LLM matchers, from the published per-token prices[2] applied to the measured input and output tokens (output includes billed thinking tokens). The underlying compute cost of serving any matcher (instance, CPU/GPU, memory) varies by deployment and is out of scope; we assume a comparable instance-serving cost across all approaches and report only the differential API cost, which the LLM matchers alone incur. The figures are therefore the additional per-decision API cost on top of a common serving baseline, not a total cost of ownership.

5 Results and Discussion

Figure 2 gives the headline: the best F_1 each matcher reaches. Tables 1 and 2 break the picture out by topic representation, so every matcher–representation pairing is visible. Because the classes are imbalanced (∼24% positive), we report precision, recall, and F_1; embeddings are scored at a cross-validated threshold and the other matchers at their native YES/NO decision, and unparseable LLM replies (at most nine per model) count as no-match.

5.1 An LLM on a description wins

The strongest matcher is an LLM reading a natural-language description. Gemini-3-Flash reaches $F_1 = 0.847$, ahead of the regex baseline (0.721) by 12.6 points and of the best embedding (0.708).

[2]Gemini API price list, accessed 2026-08-27.

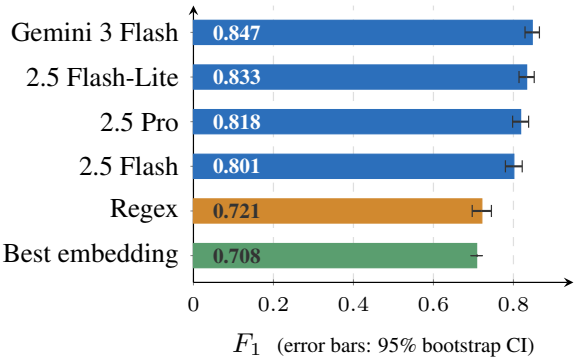


Figure 2: Main results: F_1 on the human-gold benchmark. Error bars are 95% bootstrap CIs for the native-decision methods; the **best embedding** bar is a 5-fold cross-validated point estimate (fold $\sigma = 0.027$; Table 1) and is shown without a bootstrap bar as it is not a native-decision method. **LLM matchers** (blue) lead the **regex** baseline (orange) and the best of six zero-shot embedding encoders (green); the two lightweight Flash-tier matchers top the board and the LLM–regex gap is significant ($p < 10^{-6}$ for all four matchers).

Keyphrases	P	R	F_1
Regex (baseline)	0.850	0.625	0.721
all-MiniLM-L6-v2	0.689	0.731	0.708
e5-base-v2	0.673	0.743	0.706
bge-base-en-v1.5	0.669	0.748	0.706
bge-small-en-v1.5	0.657	0.762	0.705
mpnet-base-v2	0.750	0.637	0.689
e5-small-v2	0.620	0.746	0.677
Gemini 3 Flash	0.839	0.752	**0.793**
2.5 Pro	0.845	0.725	0.780
2.5 Flash-Lite	0.882	0.638	0.740
2.5 Flash	0.898	0.603	0.721

Table 1: All matchers on the **keyphrase** representation. Embeddings are 5-fold cross-validated; regex and LLMs use their native decision. Best F_1 in bold.

The margin over regex is large and highly significant (exact McNemar $p < 10^{-6}$ for every LLM, and $p < 10^{-14}$ for the two best), and it is almost all recall: the LLM catches paraphrases and disfluent phrasings that a fixed keyphrase list misses. Surprisingly, embeddings come last though being the usual backbone of semantic matching in many other tasks.

Representation matters as much as the matcher. Tables 1 and 2 show that the best representation is not the same for every method. LLMs are strongest on the description (Gemini-3-Flash 0.847 vs. 0.793 on keyphrases; Gemini-2.5-Flash-Lite 0.833 vs. 0.740); embeddings are strongest on keyphrases (0.708 vs. 0.660); regex applies only to keyphrases,

Natural-language description	P	R	F_1
e5-base-v2	0.687	0.635	0.660
all-MiniLM-L6-v2	0.660	0.640	0.649
bge-small-en-v1.5	0.646	0.640	0.641
bge-base-en-v1.5	0.615	0.657	0.635
e5-small-v2	0.646	0.616	0.627
mpnet-base-v2	0.604	0.626	0.613
Gemini 3 Flash	0.866	0.829	**0.847**
2.5 Flash-Lite	0.887	0.786	0.833
2.5 Pro	0.827	0.809	0.818
2.5 Flash	0.931	0.703	0.801

Table 2: All matchers on the **natural-language description** representation. Regex is omitted as it matches keyphrase lists only. Embeddings are 5-fold cross-validated. Best F_1 in bold.

since a description is not a list of patterns. The optimal pairing is therefore an LLM with a natural language description, and how a topic is represented is as consequential as which matcher scores it.

A lighter model is enough. The two best systems are the two smallest LLMs, Gemini-3-Flash and Gemini-2.5-Flash-Lite, and no larger model beats them. Gemini-2.5-Flash-Lite is even nominally ahead of the heavier Gemini-2.5-Pro and Gemini-2.5-Flash ($p = 0.014/0.019$). This interesting finding suggests that a small Flash-tier model is enough for this task while heavier, most costly LLMs offer no obvious benefits.

Cost and latency. Table 3 prices the "lighter is enough" finding. Gemini-2.5-Pro is Pareto-dominated: its mandatory thinking ($\sim$525 output tokens on a `YES/NO` question) makes it the slowest (p50 5.6 s) and by far the costliest ($5.38 per 1,000 decisions), yet it does not lead on F_1. Gemini-2.5-Flash-Lite sits at the opposite corner: p50 0.35 s and $0.010 per 1,000, i.e. $\sim$16$\times$ faster and $\sim$500$\times$ cheaper than 2.5-Pro, at F_1 within 0.02 of the best matcher. Embeddings and regex are faster still and self-hosted, but trail on F_1 (Tables 1, 2). For a latency- and cost-bounded real-time deployment, a Flash-tier LLM on a description is the practical operating point.

5.2 Across three matchers

Our central finding is that LLM matchers score significantly higher on this noisy benchmark. What is clear mechanistically is that regex matches only literal keyphrases and cannot absorb disfluencies or recognition errors, and that cosine similarity rewards surface overlap rather than topic semantics when processing noisy textual input as in our setting.

Matcher	Latency (ms)		Tokens		$/1k
	p50	p95	in	out	
Gemini 3 Flash	1742	5018	100	165	0.55
2.5 Flash-Lite	346	528	100	1	0.010
2.5 Pro	5580	8365	100	525	5.38
2.5 Flash	461	771	100	1	0.033
e5-base-v2	29	32	–	–	–
all-MiniLM-L6-v2	6	7	–	–	–
Regex	0	0	–	–	–

Table 3: Per-decision latency and API cost on 200 sampled topic–utterance pairs. LLMs score the natural-language description; embedding latency is the per-utterance encode and regex the keyphrase match. Gemini $/1k is the marginal API cost (prices accessed 2026-08-27; output includes billed thinking tokens). Embeddings and regex are self-hosted and incur no API cost. Latency is measured under identical conditions from one client to the Google Gemini API; embeddings and regex are timed locally on a single CPU core.

6 Conclusion

We studied topic–utterance matching on noisy call-center ASR transcripts, comparing a regex baseline, zero-shot embeddings, and Gemini LLM matchers across two ways of describing a topic. The clear winner is an LLM reading a natural-language description: Gemini-3-Flash reaches an F_1 of 0.847, well above the regex baseline (0.721) and the best embedding ($\approx$ 0.71). Our comprehensive benchmark demonstrates that lightweight LLMs are enough for this task. The smallest Flash-tier matchers lead, while larger ones do not present superior performance. Additionally, we prove that the topic representation matters as much as the matcher: LLMs are strongest with a written description, while embeddings work better with a keyphrase list. Because the calls are sensitive customer data, we present this evaluation protocol and findings rather than the data itself. We hope this work provides insights to other practitioners working under a similar setting.

Limitations

Scope. All data are English and from a single domain, contact-center agent assist (11 companies, 173 coaching cards); we make no cross-lingual or cross-domain claims. This is an accuracy result: without a controlled clean-versus-noisy comparison, we do not isolate noise robustness from gen-

eral semantic capability.

Models. Our best matchers are proprietary Gemini API models (Gemini Team, 2025), and the very best (Gemini-3-Flash) is a preview model whose numbers are a snapshot. Our conclusions do not hinge on it: the generally available Gemini-2.5-Flash-Lite already beats every non-LLM baseline, and we anchor the comparison to a fully reproducible regex baseline and open-weight embeddings (Reimers and Gurevych, 2019). We do not test other LLM API services beyond Gemini due to our data safety policy.

Data availability. The underlying transcripts contain sensitive customer information and cannot be released. To support reproducibility within this constraint, we release the full evaluation protocol, prompts, matcher configurations, and aggregate statistics, and we report results only in aggregate; access to the data may be considered on a controlled basis subject to the service's data-processing agreements.

Ethics and Broader Impact

The data is derived from customer–agent dialogues about sensitive topics and processed in accordance with the service's data-processing and consent procedures; we use only the transcript text and do not seek to re-identify any customers or agents. We report only aggregated results, model performance scores, and evaluation methodology, not the data itself. Gold labels were obtained from internal specialized annotators who were compensated fairly.

References

Timothy Baldwin, Marie Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text (WNUT 2015), ACL-IJCNLP*.

Juan Barrionuevo-Valenzuela, Daniel Calderón-González, Zoraida Callejas, and David Griol. 2026. Supporting human operators during customer service interactions with agentic-rag. In *Proceedings of the 16th International Workshop on Spoken Dialogue System Technology (IWSDS 2026)*.

Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser. 2020. Slurp: A spoken language understanding resource package. *arXiv preprint arXiv:2011.13205*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. *arXiv preprint arXiv:2104.00783*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

Tong Cui, Jinghui Xiao, Liangyou Li, Xin Jiang, and Qun Liu. 2021. An approach to improve robustness of nlp systems against asr errors. *arXiv preprint arXiv:2103.13610*.

Jesse Davis and Mark Goadrich. 2006. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006), pp. 233-240*.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text (W-NUT 2017)*.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013)*.

Guy Feigenblat, Chulaka Gunasekara, Benjamin Sznajder, Sachindra Joshi, David Konopnicki, and Ranit Aharonov. 2021. Tweetsumm – a dialog summarization dataset for customer service. *arXiv preprint arXiv:2111.11894*.

Lingyun Feng, Jianwei Yu, Deng Cai, Songxiang Liu, Haitao Zheng, and Yan Wang. 2021. Asr-glue: A new multi-task benchmark for asr-robust natural language understanding. *arXiv preprint arXiv:2108.13048*.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin 76(5):378-382*.

Gemini Team. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Bo Han and Timothy Baldwin. 2011. Lexical normalisation of short text messages: Makn sens a #twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT 2011)*.

Matthew Honnibal and Mark Johnson. 2014. Joint incremental disfluency detection and dependency parsing. *Transactions of the Association for Computational Linguistics*, 2:131–142.

Chao-Wei Huang and Yun-Nung Chen. 2020. Learning asr-robust contextualized embeddings for spoken language understanding. *arXiv preprint arXiv:1909.10861*.

Ron Kohavi. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995), Volume 2, pp. 1137-1143*.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics 33(1):159-174*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2020. A survey on text classification: From shallow to deep learning. *arXiv preprint arXiv:2008.00364*.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2019. Benchmarking natural language understanding services for building conversational agents. *arXiv preprint arXiv:1903.05566*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*.

Mrinal Rawat and Victor Barres. 2022. Real-time caller intent detection in human-human customer support spoken conversations. *arXiv preprint arXiv:2208.06802*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Takaya Saito and Marc Rehmsmeier. 2015. The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE 10(3):e0118432*.

Ori Shapira, Shlomo E. Chazan, and Amir DN Cohen. 2025. Measuring the effect of transcription noise on downstream language understanding tasks. *arXiv preprint arXiv:2502.13645*.

Suwon Shon, Ankita Pasad, Felix Wu, Pablo Brusco, Yoav Artzi, Karen Livescu, and Kyu J. Han. 2022. Slue: New benchmark tasks for spoken language understanding evaluation on natural speech. *arXiv preprint arXiv:2111.10367*.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management 45(4):427-437*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021. Multilexnorm: A shared task on multilingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*.

Rob van der Goot and Gertjan van Noord. 2017. Monoise: Modeling noise using a modular normalization system. *arXiv preprint arXiv:1710.03476*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. 2024. C-pack: Packed resources for general chinese embeddings. *arXiv preprint arXiv:2309.07597*.

Vicky Zayats, Mari Ostendorf, and Hannaneh Hajishirzi. 2016. Disfluency detection using a bidirectional LSTM. In *Interspeech*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Junmei Zhong and William Li. 2019. Predicting customer call intent by analyzing phone call transcripts based on cnn for multi-class classification. *arXiv preprint arXiv:1907.03715*.

A Evaluation Prompt

All LLM matchers receive the identical prompt below (temperature 0). The system instruction fixes the task and the strict `YES/NO` output format; the user template is filled per example with the topic description and the customer utterance.

System instruction.

```
You are a precise judge evaluating
whether a customer utterance matches a
topic description. The topic description
defines what a customer might say when
they are discussing that topic. You must
reply with exactly one word: YES or NO.
Do not add any explanation.
```

User template.

```
Topic description:
{description}
Customer utterance:
“{utterance}”
Does the utterance match the topic?
Reply YES or NO.
```

B Annotation Guidelines

Each item shows one customer utterance and one topic (its natural-language description). Up to three annotators independently answer a single question, presented verbatim as:

> *Does the utterance fit the description? Would you say the utterance fits under the topic described?*

with response options **yes** (positive / match), **no** (negative / no-match), or **ambiguous** when the utterance is too underspecified to decide, plus an optional free-text notes field for rationale. Annotators judge *semantic* fit—whether the customer is talking about the topic—rather than surface keyword overlap, and are instructed to rely only on the utterance itself (no surrounding call context). The per-item gold label is the majority vote over the available annotators. Rows whose majority label is **ambiguous**, or with no majority (**disputed**), are excluded from the usable evaluation set; rows with only a single annotator so far are used as per-row gold but flagged and excluded from all inter-annotator agreement statistics (Section 3.2).

TurEngMix: A Text Corpus and Benchmark for Turkish-English Code-Mixed Language Identification and Named Entity Recognition

Ilayda Dogan
Computer Science
University of Maryland
idogan@umd.edu

Phuong-Anh Nguyen-Le
Information Science
University of Maryland
nlpa@umd.edu

Julia Mendelsohn
Information Science
University of Maryland
juliame@umd.edu

Abstract

Natural language processing systems underperform on code-mixed text, particularly for low-resource language pairs. Turkish-English poses a further challenge: it lets English stems combine with Turkish suffixes to form single mixed-language tokens. We introduce TurEngMix, a corpus of 5.5K noisy, naturally occurring social media posts (486,974 tokens) rich in Turkish-English code-mixing. From this corpus, we construct a new Turkish-English benchmark for code-mixed language identification (LID) and named entity recognition (NER), comprising 15K expert-annotated tokens. Evaluating both decoder LLM and fine-tuned encoder baselines, we find that monolingual Turkish and English tokens are labeled reliably, but all models have high error rates on mixed-language tokens for both LID and NER. For morphologically integrated tokens, NER error rates were 5.2× and 6.3× higher for GPT-4o and Qwen, respectively. This highlights how morphological integration remains a challenge. We release the corpus, annotations, and code to support future computational and sociolinguistic research on Turkish-English code-mixing.

1 Introduction

Code-mixing, the use of two or more languages within a single sentence or phrase, is widespread in online spaces and diaspora communities (Poplack, 1980; Androutsopoulos, 2013). It serves a range of linguistic and social functions, including identity signaling, tone-setting, and communicating culturally or technically specific concepts, and takes different structural forms across speaker backgrounds and communicative contexts (Tay, 1989). However, despite its natural role in multilingual communities and growing presence in informal, online settings, code-mixing remains underexplored in natural language processing (NLP), particularly for lower-resourced language pairs (Winata et al., 2023), reflecting the broader language diversity gap in NLP (Joshi et al., 2020; Pakray et al., 2025).

Token	Gloss
influencerlarımız	**influencer**-lar-ımız influencer- PL-1PL.POSS 'our influencers'
frameworklerinin	**framework**-ler-i- nin framework- PL-3PL.POSS-GEN 'of their frameworks'
tweetlememesi	**tweet**-le- me- me- si tweet- VBLZ-NEG-NMLZ-3SG.POSS 'the fact that they do not tweet'

Table 1: Mixed-language tokens with English stems and Turkish suffixes.

Existing code-mixing benchmarks are dominated by language pairs such as Spanish-English and Hindi-English (Doğruöz et al., 2021), where code-mixing typically occurs at word boundaries. Turkish, however, presents a markedly different linguistic setting. Its agglutinative morphology allows suffixes expressing case, possession, tense, and other grammatical functions to stack into complex words. In Turkish-English code-mixed discourse, English lexical stems are productively integrated into this structure (Yirmibeşoğlu and Eryiğit, 2018), resulting in *mixed-language tokens* that combine English roots with Turkish suffixes (examples in Table 1). These naturally occurring forms are linguistically interesting manifestations of bilingual language use that challenge traditional structural constraints of code-mixing (Kemaloglu-Er, 2018) and pose unique difficulties for otherwise simple foundational sequence-labeling tasks such as language identification (LID) and named entity recognition (NER).

These linguistic challenges are compounded by a scarcity of annotated resources. To our knowledge, there is only one publicly available Turkish-English code-mixed text dataset, comprising 391 naturally occurring examples with word-level lan-

Proceedings of the 11th Workshop on Natural User-generated Text, pages 341–355
October 28, 2026 ©2026 Association for Computational Linguistics

guage labels (Yirmibeşoğlu and Eryiğit, 2018). We are not aware of any resources for evaluating NER for Turkish-English code-mixing, nor LID with mixed-language tokens. This paper fills that gap and makes the following contributions:

- **TurEngMix**: a large-scale corpus of naturally occurring Turkish-English code-mixed social media posts, spanning a diverse range of online topics. Beyond this work, it supports future computational and sociolinguistic study of lexical innovation, code-mixing behavior, and multilingual language use.
- An annotated Turkish-English benchmark with token-level language and named-entity labels (15K tokens) derived from TurEngMix.
- A systematic evaluation of decoder LLM and fine-tuned encoder baselines for LID and NER on Turkish-English code-mixed text.

We release the corpus and annotated benchmark under a CC BY-NC-SA 4.0 license for non-commercial use. The corpus, benchmark, code, and preprocessing resources are publicly available. [1] [2]

2 Background

2.1 Turkish-English Code-Mixing

Code-mixing is well-documented among Turkish speakers in both diaspora communities and online spaces. Turkish immigrant communities in Western Europe exhibit extensive code-mixing as a norm in everyday informal in-group conversation (Backus et al., 2010), and similar patterns have been observed among Turkish-English bilinguals in the United States, where intra-sentential switching occurs at particularly high rates (Koban, 2013). Turkish-English code-mixing is also prevalent in social media contexts, where it serves a range of communicative functions including identity expression and tone-setting (Akgür, 2021).

Turkish-English written code-mixing is highly asymmetric. Turkish is agglutinative, verb-final, and suffixal. Following the Matrix Language Frame Model (Myers-Scotton, 1993), this makes it natural for Turkish to provide the matrix frame and English to supply content stems that take Turkish inflection (e.g., English nouns and adjectives bearing Turkish case, possessive, or derivational suffixes). Mixed-language tokens (i.e., English roots with Turkish suffix chains) are frequent in informal written communication, such as in Turkish social media data (Yirmibeşoğlu and Eryiğit, 2018). Accurately processing such tokens is thus an essential prerequisite for downstream NLP modeling and analysis in this domain, but remains a challenge for extant systems.

2.2 Code-Mixing Resources and Evaluation

Most existing code-mixed NLP resources are concentrated on a small set of language pairs, mainly Spanish-English and Hindi-English (Doğruöz et al., 2021; Winata et al., 2023). Central benchmarks (e.g., LinCE (Aguilar et al., 2020) and GLUECoS (Khanuja et al., 2020)) explicitly acknowledge this skew as they provide evaluation limited to a few popular pairs. This imbalance is a core structural problem in the field (Doğruöz et al., 2023). More recent work has begun expanding corpus coverage to underrepresented language pairs. For example, VietMix (Tran et al., 2026) introduced an expert-translated Vietnamese-English corpus for machine translation, demonstrating both the utility of carefully curated resources and the continued difficulty that modern models face on code-mixed text. Nevertheless, similar resources remain unavailable for many morphologically and typologically distinct language pairs, including Turkish-English.

While performance on lower-level tasks such as LID and NER has improved for code-mixed data (Winata et al., 2023; Pakray et al., 2025; Sheth et al., 2025), these tasks remain challenging in realistic multilingual settings. Recent studies have shown that LID performance can degrade across domains (van der Goot, 2025), and code-switched LID remains difficult even across multiple language pairs, with existing approaches struggling to reliably identify language boundaries (Burchell et al., 2024). Similarly, multilingual NER benchmarks continue to expose challenges involving noisy text, fine-grained entity categories, and complex entity structures (Malmasi et al., 2022). Recent work has also explored specialized approaches for code-mixed LID, including training-free methods such as MaskLID (Kargaran et al., 2024). However, there remains a gap between higher-resourced and lower-resourced language pairs. Moreover, performance on more complex NLP tasks such as sentiment analysis and question answering still lags behind that of non-mixed data. For example, CodeMixBench, a multilingual benchmark spanning eighteen languages (though no Turkic languages), reports persistent performance degradation on code-mixed in-

[1] https://huggingface.co/datasets/ilydoa/TurEngMix
[2] https://github.com/kemnguyenle/TurEngMix

uı ux designer and front end developer olarak uygun fiyatlı **b2c b2b**, wordpress, e-ticaret gibi çeşitli alanlarda **web siteleri** yapabilirim. ulaşmak isteyenler yeşillendirsin.
tanım : sertbest çalışan

Topic Tag: freelance

Figure 1: Example *Ekşi Sözlük* post collected for the TurEngMix corpus, with code-mixing (English words mixed with Turkish text) highlighted for reference.

puts across a range of NLP tasks despite recent advances in LLMs (Yang and Chai, 2025).

There are some resources for tasks such as sentiment analysis and question-answering in Turkish (Çöltekin et al., 2023), and transformer models trained specifically for Turkish (Schmitt and Schweter, 2026; Uludoğan et al., 2024). For code-mixing, there exists a publicly available Turkish-German Twitter/X corpus (Çetinoğlu, 2016) and an audio corpus (Çetinoğlu, 2017). However, we are only aware of one Turkish-English code-switched written corpus, which comprises 391 naturally-occurring samples collected from social media settings and is designed for the task of code-switching detection (Yirmibeşoğlu and Eryiğit, 2018). To our knowledge, TurEngMix is the first Turkish–English benchmark jointly annotated for named entity recognition and token-level language identification on naturally occurring forum text, and the largest such resource to include explicit metadata for English-origin stems with Turkish morphology.

3 TurEngMix Corpus

We first introduce the data collection and filtering steps to create the TurEngMix corpus and present corpus statistics. In the next section, we describe our annotated benchmark for LID and NER, followed by our evaluation of baseline models.

3.1 Data Collection

The corpus consists of public Turkish social media posts from the Turkish-language social media site *Ekşi Sözlük*, a Reddit-like social media site with posts organized by topic tags, which function similarly to subreddits. The filtered dataset, token-level annotations, and preprocessing scripts are publicly available.

We collected 30K publicly-available posts using the `eksipy` API wrapper. Because posts could be collected by their topic tag category, we first curated a diverse list of topic tags that were likely to surface code-mixing, including tech topics, social media/entertainment, lifestyle, education, and more (see Appendix for the full topic list). The posts were retrieved as text strings. No usernames, profile information, timestamps, or other personally identifying metadata were retained.

Statistic	Value
Unique posts	5,549
Total tokens	486,974
Post length range	3–500 words
Mean post length	87.75 words
Median post length	57 words

Table 2: Summary statistics of the TurEngMix corpus.

3.2 Data Filtering

To increase the prevalence of Turkish-English code-mixing, we applied a three-stage filtering pipeline. First, the *langdetect* Python library was used to identify posts containing both Turkish and English. We then applied two GPT-4o filtering stages: the first performed a binary check for the presence of English lexical material, while the second repeated this while excluding posts in which apparent English content consisted solely of URLs, brand names, or other non-linguistic strings that could produce false positives. Lastly, posts containing fewer than three words or more than 500 words, where a word is defined as a whitespace-separated character sequence, were excluded after filtering.

To assess the filtering pipeline, we manually inspected 100 retained and 100 discarded posts. Of the retained posts, 71 contained Turkish-English code-mixing, compared with only 22 of the discarded posts, indicating that filtering substantially raised the density of code-mixed content relative to the unfiltered pool. Not every retained post is code-mixed, and some code-mixed posts are discarded. The retained set, therefore, is strongly enriched for code-mixing rather than exhaustive. The remaining monolingual posts serve as useful negative examples for downstream LID and NER analyses.

3.3 Corpus Statistics

The final TurEngMix Corpus consists of 5,549 unique posts comprising 486,974 tokens (Table 2). Posts have a mean length of 87.75 words and a median length of 57 words. Figure 2 shows the distribution of post lengths, illustrating that while most posts are relatively short, the corpus also captures longer-form text posts characteristic of online

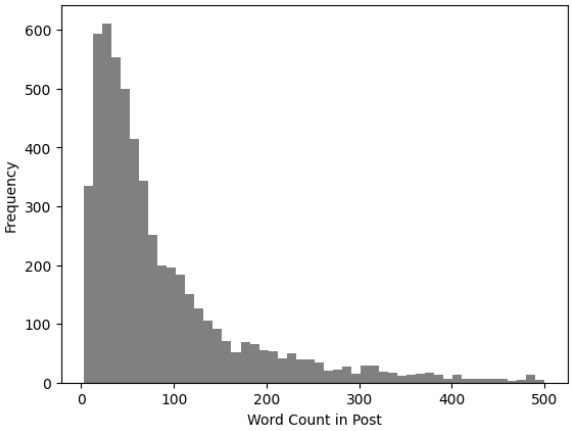


Figure 2: Word count distribution in the final dataset. Post word count was majority within 100 words, with longer posts scaling to 500.

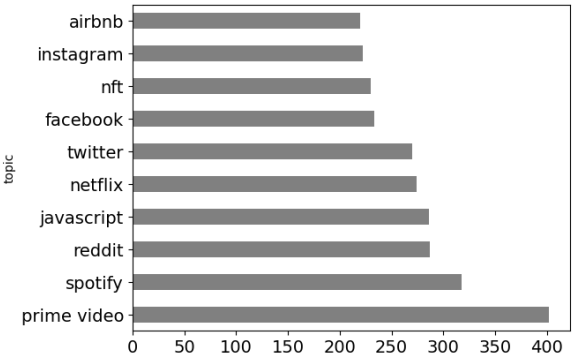


Figure 3: Ten most-frequent topic tags in the final dataset are primarily tech and entertainment.

forums.

The corpus spans 71 unique topic tags. Technology and social media related topics (e.g., *twitter*, *netflix*, and *javascript*) are among the most frequently represented categories (Figure 3). The prominence of technology and social media topics is consistent with the documented dominance of English in online and technical domains (Danet and Herring, 2007), which makes such settings especially conducive to code-mixing (Androutsopoulos, 2013; Akgür, 2021).

The corpus captures a broad spectrum of code-mixing behavior, including isolated English lexical insertions, longer mixed-language phrases, and morphologically integrated English stems with Turkish suffixes (Table 3). As the TurEngMix corpus preserves naturally occurring Turkish-English discourse across diverse online contexts, it supports future investigations of code-mixing behavior, multilingual communication, and language variation, while also providing a resource for developing and evaluating NLP systems beyond LID and NER.

4 LID and NER Annotated Benchmark

To enable standardized evaluation of foundational sequence labeling tasks on Turkish-English code-mixing, we constructed an annotated benchmark comprising 15,012 annotated tokens from the TurEngMix Corpus. By deriving the benchmark directly from naturally occurring social media posts, the annotated dataset preserves the linguistic characteristics of authentic Turkish-English code-mixed discourse.

4.1 Benchmark Construction

We first randomly sampled 250 posts from the corpus, then tokenized each post with a custom Python script that separates text based on whitespace and punctuation boundaries while preserving alphanumeric strings and morphologically mixed tokens. We removed hyperlinks, non-linguistic symbols, and special characters that did not contribute to language identification or named entity annotation. Punctuation marks such as apostrophes or quotation marks were retained because of their role in building morphologically complex words. The tokenization script and annotation guidelines are publicly available alongside the released dataset.

4.1.1 Annotation Scheme

Each token received a language identification label and a named entity label, as shown in Figure 4.

LID We adopted the annotation framework of Solorio et al. (2014), labeling each token as TR, EN, MIXED, OTHER, AMBIGUOUS, or NE. Following the shared task convention, all named entities were assigned the NE label regardless of language of origin or the presence of Turkish suffixes. For example, *netflix'in* was labeled NE, not MIXED, despite containing a Turkish possessive suffix.

NER For NER, we followed the CALCS 2018 shared task guidelines (Aguilar et al., 2018). Named entities were annotated as Person, Organization, Location, Group, Product, Title, Event, Time, or Other using the BIO tagging scheme.

Code-Mixing Metadata We also annotated for a broader "INTEGRATED" token category which captured whether an English-origin word, including named entities and brands, had a Turkish suffix. There were 386 tokens in this category, including

Code-mixing type	Original post	English translation
Isolated English insertions	her açtığımda belki gülhan şen kaydolmuştur diye **search** edip havamı aldığım **trendy** oluşum.	The *trendy* platform that, every time I open it, I *search* thinking 'maybe Gülhan Şen has signed up' and end up being disappointed.
Mixed-language phrase	**love is an excuse to get hurt** diye çok mantıklı bir açıklamasını yapmış birileri	Someone gave a very reasonable explanation that **"love is an excuse to get hurt."**
Mixed-language token	Bu **link'teki** video mutlaka izlenmeli bence.	The video at **this link** should definitely be watched.

Table 3: Different types of code-mixing found in the dataset, including isolated English insertions, longer mixed language phrases, and mixed-language tokens.

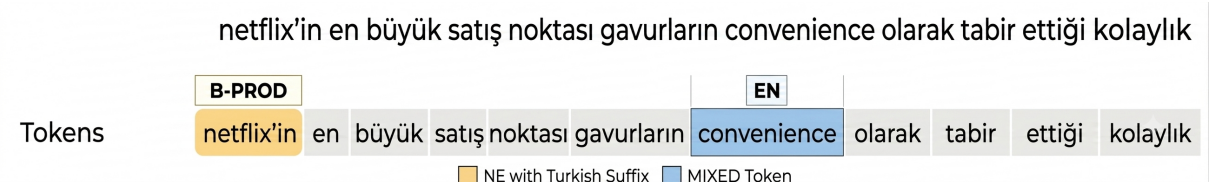


Figure 4: An example annotation sentence from the golden annotated dataset. Translation: *Netflix's biggest selling point is the ease of use which the Westerners refer to as 'convenience.'*

tokens that had been labeled as NE in LID annotations but contained morphological mixing. This allowed us to understand the prevalence of the tokens that displayed the morphological mixing behavior.

Posts were also labeled with the type of code-mixing present (see Table 3), including INTEGRATED tokens embedded into the post, isolated English words inserted into Turkish text, and English phrases (> 2 words) present in the text. There was overlap between categories.

4.1.2 Annotation Process

The first author, fluent in both Turkish and English, annotated each token. To validate, inter-annotator agreement was calculated with a secondary annotator (also fluent in both languages) on a sample of 203 tokens from randomly sampled posts. A Cohen's Kappa score of 0.96 was achieved for LID and 0.94 for NER. Per-class agreement was high across categories (see Appendix A.3), with the lowest on MIXED ($\kappa = 0.75$).

Disagreements primarily involved lexicalized English loanwords (e.g., *videolar*, which may reasonably be interpreted as either Turkish or morphologically mixed) and distinctions between semantically related named entity categories (e.g., Product versus software-related terms). These cases were resolved through discussion, yielding complete agreement in the final annotations.

LID Label	Count
TR	11,723
EN	1,992
NE	1042
Mixed	189
Ambiguous	62
Other Language	4
Total	15,012

Table 4: Distribution of LID labels. Note that NE tokens included additional morphologically mixed examples.

4.2 Benchmark Statistics

The benchmark exhibits substantial code-mixing, with a mean document-level Code-Mixing Index (CMI) of 13.25 (SD = 13.12) and 87.2% of posts containing more than one language (Das and Gambäck, 2014). Although morphologically integrated tokens account for only 2.57% of all annotated tokens, they appear in 66.0% of posts, indicating that Turkish morphological integration is a frequent characteristic of code-mixed discourse despite its relatively low token frequency. Beyond morphologically integrated tokens, the benchmark also captures other common forms of Turkish-English code-mixing. A total of 154 posts contain isolated English insertions, while 148 contain longer embedded English phrases (Table 6).

For the encoder models, the benchmark was partitioned into training, validation, and test splits using an 80/10/10 ratio, with posts (rather than individual tokens) assigned to splits to prevent in-

NER Label	Count
Non-Entity (O)	13,971
Product	359
Title	246
Organization	129
Person	119
Location	88
Other	52
Group	26
Event	19
Time	3
Total	**15,012**

Table 5: Distribution of named entity labels.

Split	Posts
Embedded Mixed Token	165
Isolated English Token	154
Embedded English Phrase	148

Table 6: Distribution of code-mixing patterns in the benchmark.

formation leakage across sets (Table 7). Each split maintained a similar distribution of LID and NER labels, with most labels represented (See Table 4 and 5 for benchmark label distributions). For NER, some of the rarest NE labels (Event, Time, etc) were not represented in the test set.

5 Baseline Evaluations

We present baseline evaluations of pretrained encoder and decoder models on two mature sequence-labeling tasks—LID and NER.

5.1 Decoder Setup

We selected GPT-4o and Qwen3-8B to represent two complementary classes of contemporary LLMs: a large proprietary model and a compact open-weight multilingual model.

For both tasks, models received complete posts reconstructed from the benchmark as sequences of pre-tokenized, indexed tokens, rather than receiving isolated words. This preserved the original sentential context of each social media post and allowed models to leverage context when predicting on ambiguous tokens. Models were prompted to assign a label to each pre-tokenized input token while preserving its provided token index. For LID, labels corresponded to the benchmark language categories. For NER, models predicted BIO-formatted named entity labels. Predicted labels were aligned with the benchmark using the provided token indices prior to scoring, and malformed outputs were

Split	Posts	Sentences	Tokens
Train	200	254	12,466
Validation	25	26	1,173
Test	25	41	1,373
Total	250	321	15,012

Table 7: Dataset statistics across train, validation, and test splits.

treated as incorrect.

Model temperature was fixed at 0 to encourage deterministic outputs, and prompts required the model to assign exactly one label to each token. We evaluated both zero-shot and three-shot prompting to measure the effect of in-context examples. The three demonstration examples were selected to cover the range of language and named entity labels present in the benchmark. Complete prompts are included in the Appendix.

Predictions were produced for the complete benchmark (250 posts) and compared against the gold-standard annotations. We report macro-averaged F1 scores together with per-label F1 scores for both the LID and NER tasks.

5.2 Encoder Setup

To compare general-purpose LLMs with specialized encoder architectures, we evaluated three pretrained transformer encoders: XLM-RoBERTa (Conneau et al., 2020), BERTurk (Yildirim, 2024), and TurkishBERTweet (Najafi and Varol, 2023). These models represent different sources of linguistic specialization: multilingual representation learning, general Turkish language modeling, and social-media-specific Turkish language modeling.

Models were fine-tuned for token classification using the Hugging Face Transformers framework on the TurEngMix Benchmark training split, with a majority-class baseline reported for comparison. Because encoder tokenizers operate on subword units, benchmark tokens were further segmented using each model's tokenizer, with labels assigned only to the first subword piece and continuation subwords ignored during training and evaluation.

Fine-tuning was performed using the Hugging Face Trainer with a learning rate of 2×10^{-5}, weight decay of 0.01, and a maximum of 10 epochs. Each model was trained using four random seeds. For each run, the checkpoint with the highest validation F1 score was evaluated on the held-out test split, and we report the mean and standard deviation across runs. Predictions were evaluated

against the gold-standard annotations using macro-averaged and per-label F1 scores.

6 Results

6.1 Language Identification

Distinguishing morphologically integrated English borrowings from ordinary Turkish or English tokens is the central difficulty in Turkish-English LID. Overall macro-F1 is modest across all models (Table 9): GPT-4o is strongest, with three-shot prompting giving a small gain over zero-shot (0.5170→0.5238), while Qwen3-8B does not benefit from demonstrations (0.4920→0.4788). Fine-tuned encoders are comparable overall, with BERTurk and TurkishBERTweet leading among them. The aggregate scores, however, mask sharp per-class variation (Table 8): every model labels Turkish and English tokens well (F1 > 0.84), but MIXED tokens and named entities are far harder. GPT-4o reaches only 0.45 F1 on MIXED tokens; the best encoder (TurkishBERTweet) reaches 0.22. Encoders match the LLMs on named entities (F1 0.56–0.68) but the results are much lower on MIXED, confirming that morphological integration—not language identity per se—is what current models struggle with (Table 8).

6.2 Named Entity Recognition

Table 10 reports overall BIO-level NER performance. GPT-4o outperformed Qwen3-8B under both prompting settings, and three-shot prompting improved GPT-4o from a macro F1 of 0.5669 to 0.6140. In contrast, encoder models achieved considerably lower macro F1 despite high token-level accuracy, reflecting the strong class imbalance in the dataset where non-entity tokens account for approximately 93% of all tokens.

Because BIO evaluation jointly measures entity detection, entity classification, and boundary prediction, we additionally evaluate binary entity detection by collapsing all entity categories into entity (NE) and non-entity (Non-NE) (Table 11). GPT-4o achieved an F1 of 0.80 on the NE class under three-shot prompting, suggesting that many BIO-level errors were associated with correctly identifying the specific entity type or boundary prediction rather than failure to detect named entities. Qwen achieved a lower F1 of 0.70 on the NE class, while encoder models ranged from 0.47 to 0.67.

Per-category evaluation reveals wide variation across entity types. Person, Location, and Title entities generally had higher F1 scores, whereas Organization, Other, Event and Time were substantially lower (Appendix Tables A3-A4). Across all evaluated models, Organization and Product were the most frequently confused categories, particularly for technology-related entities whose interpretation depended on context (e.g., *Google*, *Netflix*, and *Microsoft*).

6.3 Error Analysis

Qualitative inspection demonstrates consistent error patterns across both sequence labeling tasks.

6.3.1 LID

For language identification, both GPT-4o and Qwen frequently misclassified morphologically integrated English borrowings as ordinary Turkish tokens, particularly when multiple Turkish suffixes were attached (e.g., *entryden*, *influencerlarımız*, and *editleyip*). Named entities containing Turkish suffixes (e.g., *Microsoft'un* and *Sony'nin*) were also commonly predicted as MIXED rather than NE, suggesting that models relied heavily on surface morphology instead of treating named entities as a separate language category. Conversely, English compounds containing hyphens (e.g., *k-food* and *marketing-mix*) were often incorrectly labeled as MIXED despite containing no Turkish morphology.

6.3.2 NER

For named entity recognition, the most common errors involved morphologically integrated English borrowings and technology-related named entities. Technology entities frequently produced Product-Organization confusion (e.g., *Twitter*, *Google*, *Cloudflare*, and *Last.fm*), while morphologically integrated entities such as *reactlerle* and *python'daki* were often missed after Turkish suffix attachment.

6.3.3 Morphological Integration in Named Entities

To further examine whether morphological integration affects NER performance, we conducted an error analysis on the GPT-4o three-shot and Qwen three-shot settings. We compared NER error rates between tokens containing English-origin stems bearing Turkish suffixes (including morphologically adapted named entities) and all other evaluated tokens (Table 12). GPT-4o produced errors on 19.7% of mixed tokens compared with only 3.8% of other tokens. A chi-square test confirmed that this difference was highly significant ($\chi^2 = 231.83$, $p < 0.001$), corresponding to an

Model	Setting	TR	EN	MIXED	NE	OTHER	AMBIGUOUS
GPT-4o	zero-shot	0.93	0.89	0.42	0.62	**0.19**	0.00
GPT-4o	three-shot	**0.98**	**0.90**	**0.45**	0.67	0.13	0.00
Qwen3-8B	zero-shot	**0.98**	0.86	0.33	0.60	0.07	**0.12**
Qwen3-8B	three-shot	**0.98**	0.86	0.39	0.54	0.10	0.00
XLM-RoBERTa	fine-tuned	**0.98**	0.87	0.00	**0.68**	0.00	0.00
BERTurk	fine-tuned	0.97	0.84	0.13	0.64	0.00	0.00
TurkishBERTweet	fine-tuned	**0.98**	0.84	0.22	0.56	0.00	0.00

Table 8: Per-class F1 scores for LID. Decoder LLMs were tested on the full dataset. Encoder models were fine-tuned on the train/dev and evaluated on the test set.

Model	Setting	Macro F1
GPT-4o	Zero-shot	0.5170
GPT-4o	Three-shot	**0.5238**
Qwen3-8B	Zero-shot	0.4920
Qwen3-8B	Three-shot	0.4788
Baseline	–	0.1414
XLM-RoBERTa	Fine-tuned	0.4358 ± 0.0137
BERTurk	Fine-tuned	**0.4734 ± 0.0355**
TurkishBERTweet	Fine-tuned	0.4195 ± 0.0177

Table 9: Macro F1 scores for LID task. Decoder LLMs were tested on the full dataset. Encoders were fine-tuned on the train/dev and evaluated on the test set.

Model	Setting	Macro F1
GPT-4o	Zero-shot	0.5669
GPT-4o	Three-shot	**0.6140**
Qwen3-8B	Zero-shot	0.3993
Qwen3-8B	Three-shot	0.4162
Baseline	–	0.0502
XLM-RoBERTa	Fine-tuned	0.0772 ± 0.0246
BERTurk	Fine-tuned	**0.1759 ± 0.0550**
TurkishBERTweet	Fine-tuned	0.0821 ± 0.0273

Table 10: BIO-level NER macro F1.

odds ratio of 6.21. Qwen exhibited an even larger disparity, with error rates of 33.5% and 5.3%, respectively ($\chi^2 = 507.17$, $p < 0.001$; odds ratio = 8.84). These results suggest that morphological integration introduces an additional challenge beyond general code-mixed language processing.

7 Conclusion

This work addresses gaps in Turkish-English code-mixed NLP by introducing a large-scale corpus, an expert-annotated benchmark for language identification and named entity recognition, and systematic evaluations of current LLM and encoder-based models. Our analysis reveals that morphologically mixed tokens remain a persistent challenge for multilingual NLP systems, highlighting an underexplored difficulty for agglutinative languages

Model	Setting	Non-NE	NE
GPT-4o	Zero-shot	**0.98**	0.78
GPT-4o	Three-shot	**0.98**	**0.80**
Qwen3-8B	Zero-shot	0.97	0.70
Qwen3-8B	Three-shot	**0.98**	0.73
XLM-RoBERTa	Fine-tuned	0.97	**0.67**
BERTurk	Fine-tuned	0.96	0.47
TurkishBERTweet	Fine-tuned	0.97	0.62

Table 11: F1 scores for binary NER. BIO labels are collapsed into two categories: NE and NON-NE.

Model	Other Tokens	Mixed Tokens
GPT-4o	3.8%	19.7%
Qwen3-8B	5.3%	33.5%

Table 12: Comparison of NER error rates between morphologically integrated English borrowings and other evaluated tokens.

in code-mixed settings. These findings suggest that progress on code-mixed NLP for agglutinative languages may require morphology-aware approaches, such as sub-token language attribution or morphological segmentation as prepocessing, rather than scale alone. By releasing these resources and evaluations, we aim to support future research on multilingual language understanding and improve the representation of diverse language varieties in NLP.

8 Limitations

As the data was collected from *Ekşi Sözlük*, a social media site with a specific user base, this Turkish-English code-mixing resource is thus only most accurately representative of the code-mixing practiced by that online community, and does not account for different patterns in other domains. Furthermore, because topic tags likely to contain Turkish-English code-mixing were prioritized during corpus construction, the corpus is not intended to estimate the prevalence of code-mixing in Turkish online communication. As the site is public,

pretraining contamination is possible. Future work could expand to other online communities or other settings (e.g. spoken, in-person).

A further consideration is that GPT-4o served both as a filtering component during corpus construction (§3.2) and as an evaluated baseline. Posts in the corpus are, by construction, posts in which GPT-4o detected English lexical material, which may favor GPT-4o's LID performance relative to models that played no role in dataset creation. We partially mitigate this by including a langdetect-based first stage and by manual inspection of retained and discarded posts, but results for GPT-4o should be interpreted with this selection effect.

The annotation scheme follows established task definitions, but the separate NE category can obscure cases where named entities also exhibit Turkish morphological integration; we partially address this by separately annotating the "INTEGRATED" category for all English-origin tokens with Turkish morphological suffixes. The LLM prompts also did not systematically provide demonstrations for every annotation category, leaving the effect of alternative prompting strategies as an important direction for future work.

Another limitation was the class imbalance present in the dataset. Some LID and NER categories contained only a small number of instances, limiting the reliability of per-category evaluation for rare labels. For LID, the majority of tokens were TR tokens. For NER, O (non-entity) tokens comprised the majority of the dataset.

References

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Mona Diab, Julia Hirschberg, and Thamar Solorio. 2018. Named entity recognition on code-switched data: Overview of the CALCS 2018 shared task. In *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*, pages 138–147, Melbourne, Australia. Association for Computational Linguistics.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Sevilen Akgür. 2021. A case study of turkish-english written codeswitching on twitter with specific reference to language awareness. Master's thesis, Eastern Mediterranean University (EMU)-Doğu Akdeniz Üniversitesi (DAÜ).

Jannis Androutsopoulos. 2013. Code-switching in computer-mediated communication. *Pragmatics of computer-mediated communication*, 9:667–694.

A. Backus, J.N. Jörgensen, and C.W. Pfaff. 2010. Linguistic effects of immigration: Language choice, codeswitching, and change in western european turkish. *Language & Linguistics Compass*, 4(7):481–495.

Laurie Burchell, Alexandra Birch, Robert Thompson, and Kenneth Heafield. 2024. Code-switched language identification is harder than you think. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 646–658, St. Julian's, Malta. Association for Computational Linguistics.

Özlem Çetinoğlu. 2016. A Turkish-German code-switching corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4215–4220, Portorož, Slovenia. European Language Resources Association (ELRA).

Özlem Çetinoğlu. 2017. A code-switching corpus of Turkish-German conversations. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.

Çağrı Çöltekin, A Seza Doğruöz, and Özlem Çetinoğlu. 2023. Resources for turkish natural language processing: A critical survey. *Language resources and evaluation*, 57(1):449–488.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Brenda Danet and Susan C. Herring, editors. 2007. *The Multilingual Internet: Language, Culture, and Communication Online*. Oxford University Press, Oxford ; New York.

Amitava Das and Björn Gambäck. 2014. Identifying languages at the word level in code-mixed Indian social media text. In *Proceedings of the 11th International Conference on Natural Language Processing*, pages 378–387, Goa, India. NLP Association of India.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *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), pages 1654–1666, Online. Association for Computational Linguistics.

A. Seza Doğruöz, Sunayana Sitaram, and Zheng Xin Yong. 2023. Representativeness as a forgotten lesson for multilingual and code-switched data collection and preparation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5751–5767, Singapore. Association for Computational Linguistics.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Amir Hossein Kargaran, François Yvon, and Hinrich Schuetze. 2024. MaskLID: Code-switching language identification through iterative masking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 459–469, Bangkok, Thailand. Association for Computational Linguistics.

Elif Kemaloglu-Er. 2018. Patterns of intrasentential code-switching in turkish-english bilingual discourse: Testing the free morpheme and the equivalence constraint. *Artıbilim: Adana Bilim ve Teknoloji Üniversitesi Sosyal Bilimler Dergisi*, 1(2):35–45.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Didem Koban. 2013. Intra-sentential and inter-sentential code-switching in turkish-english bilinguals in new york city, us. *Procedia-Social and Behavioral Sciences*, 70:1174–1179.

Shervin Malmasi, Anjie Fang, Besnik Fetahu, Sudipta Kar, and Oleg Rokhlenko. 2022. SemEval-2022 task 11: Multilingual complex named entity recognition (MultiCoNER). In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1412–1437, Seattle, United States. Association for Computational Linguistics.

Carol Myers-Scotton. 1993. *Duelling Languages: Grammatical Structure in Codeswitching*. Oxford University Press, New York.

Ali Najafi and Onur Varol. 2023. Turkishbertweet: Fast and reliable large language model for social media analysis. *Preprint*, arXiv:2311.18063.

Partha Pakray, Alexander Gelbukh, and Sivaji Bandyopadhyay. 2025. Natural language processing applications for low-resource languages. *Natural Language Processing*, 31(2):183–197.

Shana Poplack. 1980. Sometimes i'll start a sentence in spanish y termino en espaÑol: toward a typology of code-switching 1. *Linguistics*, 18:581–618.

Raphael Schmitt and Stefan Schweter. 2026. Sindbert, the sailor: Charting the seas of turkish nlp. In *Proceedings of the Second Workshop Natural Language Processing for Turkic Languages (SIGTURK 2026)*, page 1–13. Association for Computational Linguistics.

Rajvee Sheth, Himanshu Beniwal, and Mayank Singh. 2025. COMI-LINGUA: Expert annotated large-scale dataset for multitask NLP in Hindi-English code-mixing. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7973–7992, Suzhou, China. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

Mary WJ Tay. 1989. Code switching and code mixing as a communicative strategy in multilingual discourse. *World Englishes*, 8(3):407–417.

Hieu Tran, Phuong-Anh Nguyen-Le, Huy Nghiem, Quang-Nhan Nguyen, Wei Ai, and Marine Carpuat. 2026. VietMix: A naturally-occurring parallel corpus and augmentation framework for Vietnamese-English code-mixed machine translation. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7266–7284, Rabat, Morocco. Association for Computational Linguistics.

Gökçe Uludoğan, Zeynep Balal, Furkan Akkurt, Meliksah Turker, Onur Gungor, and Susan Üsküdarlı. 2024. TURNA: A Turkish encoder-decoder language model for enhanced understanding and generation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10103–10117, Bangkok, Thailand. Association for Computational Linguistics.

Rob van der Goot. 2025. Identifying open challenges in language identification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18207–18227, Vienna, Austria. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Zheng-Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in nlp: A systematic survey on trends and challenges. *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978.

Yilun Yang and Yekun Chai. 2025. Codemixbench: Evaluating code-mixing capabilities of llms across 18 languages. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2139–2169.

Savas Yildirim. 2024. Fine-tuning transformer-based encoder for turkish language understanding tasks. *Preprint*, arXiv:2401.17396.

Zeynep Yirmibeşoğlu and Gülşen Eryiğit. 2018. Detecting code-switching between Turkish-English language pair. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 110–115, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

A.1 Dataset Collection

The TurEngMix Corpus was collected from publicly available posts on *Ekşi Sözlük* during November 2025. Candidate posts were retrieved using topic tags selected to increase the likelihood of Turkish-English code-mixing.

A.1.1 Topic Tags

```
topic_names = [
    #Tech / Programming / AI
    "spotify", "google", "python", "ai",
    "chatgpt", "openai", "midjourney", "dalle",
    "javascript", "react", "nodejs", "git",
    "blockchain", "web3", "crypto", "nft",
    "startup", "venture capital",

    #Social Media / Internet Culture
    "facebook", "instagram", "twitter",
    "youtube", "tiktok", "linkedin", "discord",
    "reddit",
    "influencer", "meme", "shitpost", "gaming",
    "twitch", "streamer",

    #Entertainment / Gaming / Anime
    "netflix", "prime video", "disney+",
    "minecraft", "lol", "anime", "kpop",
    "manga", "webtoon",
    "rap", "hiphop", "edm", "dj",

    #Sports
    "futbol", "basketbol", "tennis", "formula
    1", "voleybol", "gym", "workout", "fitness",

    #Lifestyle / Travel / Food / Shopping
    "yemek", "tatil", "seyahat", "alışveriş",
    "fashion", "sneakers", "nike", "adidas",
    "apple", "samsung", "mobile", "app",
    "airbnb", "hotel", "vegan", "fitness
    lifestyle",

    #Daily Life / Work / Education
    "üniversite", "iş", "freelance",
    "marketing", "growth", "seo", "branding",
    "startup life",
    "gündem", "aşk", "sağlık", "covid",
    "ekonomi", "politika", "haberler",
    "finance",

    #Misc / Fun / Pop Culture
    "lol memes", "viral", "challenge", "trend",
    "gaming memes", "internet slang",
    "funny videos", "edutainment", "tech news",
    "apps review", "music", "concert",
    "festival"
]
```

A.2 Corpus Filtering

Corpus construction employed a multi-stage filtering pipeline to identify posts containing Turkish-English code-mixing while removing noise.

A.2.1 Stage 1 Prompt: English Word Detection

Initial filtering prompt used to identify posts containing English words.

```
prompt_text = """
Detect if the given Turkish social media post
    contains any English words. If English
    words are present, return true.

Steps
1. Language Detection: Detect whether the given
    text contains English words.

Output Format:
Detection Result: Output "True" if English
    words are present, otherwise "False".
    Number your answers to match the post
    numbers, and put each response on its own
    line.

       Example output:
       1. True
       2. False
       etc
"""
```

A.2.2 Stage 2 Prompt: Code-Mixing Verification

Second-stage filtering prompt providing a more explicit instructions for detection of Turkish-English code-mixing.

```
prompt_text = """
You will be given Turkish social media posts.
    Go through the entire post, word by word,
    and detect if it contains any English words
    mixed into the Turkish, including
    individual English words or
    sentences/phrases in English. Don't count
    URLs or brand names as English words.

If English words are present in the post,
    return true.

Steps
1. Language Detection: Detect whether the given
    post contains English words.

Output Format:
Detection Result: Output "True" if English
    words are present, otherwise "False".
Number your answers to match the post numbers,
    and put each response on its own line.

       Example output:
       1. True
       2. False
       etc
"""
```

A.2.3 Language Identification Prompt

Prompt template used for LLM-based language identification experiments.

```
prompt_text = """
Determine the language label for each token in
    the input text.

Use the following labels:
TR: Turkish
EN: English
MIXED: Mixed Turkish and English
OTHER: Any other language
NE: Named entity (representing all or part of a
    named entity)
AMBIGUOUS: Could belong to either language in
    context

The input consists of numbered tokens. Each
    input line contains an integer index,
    followed by a tab character, followed by a
    token.

For every input line:
- Output exactly one line.
- Preserve the same INDEX.
- Assign exactly one language label.
- Do not skip any indices.
- Do not duplicate any indices.
- Do not renumber the indices.

Output each line in the following format:

INDEX|LABEL

The number of output lines must exactly equal
    the number of input lines.

Return only the INDEX|LABEL pairs, one per
    line. Do not include the original tokens,
    explanations, numbering, headings, blank
    lines, or any other text.
"""
```

A.2.4 Named Entity Recognition Prompt

Prompt template used for LLM-based named entity recognition experiments.

```
prompt_text = """
Analyze the following text and identify named
    entities using the BIO tagging scheme.

Use the following BIO format:
- B-: The first token of a named entity.
- I-: Any subsequent token belonging to the
    same named entity.
- O: A token that is not part of any named
    entity.

Use the following entity categories:

- B-PER / I-PER (Person): Proper names or
    nicknames that uniquely identify a real or
    fictional person, including artists and
    famous people.
- B-ORG / I-ORG (Organization): Companies,
    institutions, corporations, and other
    entities that have members or employees and
    act as a whole. When an entity could refer
    to either an organization or another
    category (e.g., Facebook), use the
```

```
    surrounding context to determine the
    correct label.
- B-LOC / I-LOC (Location): Physical places
    that people can visit, including countries,
    cities, addresses, facilities, landmarks,
    restaurants, and tourist attractions. When
    an organization name is used to refer to a
    physical place, label it as LOC.
- B-GROUP / I-GROUP (Group): Sports teams,
    music bands, duos, and similar groups of
    people.
- B-PROD / I-PROD (Product): Manufactured
    products, medicines, food products,
    well-defined services, website
    applications, website accounts, devices,
    and similar products. Distinguish products
    from organizations using context.
- B-TITLE / I-TITLE (Title): Titles of creative
    works such as movies, books, TV shows,
    songs, and similar media.
- B-EVENT / I-EVENT (Event): Named events such
    as concerts, competitions, conferences,
    festivals, and award ceremonies. Do not use
    this category for holidays.
- B-TIME / I-TIME (Time): Months, weekdays,
    seasons, holidays, and recurring calendar
    dates. Do not label hours, minutes,
    seconds, or relative temporal expressions
    such as "yesterday," "tomorrow," "week," or
    "year."
- B-OTHER / I-OTHER (Other): Named entities
    that do not fit the categories above,
    including nationalities, languages, music
    genres, and similar entities.
- O (None): Not part of a named entity

The input consists of numbered tokens. Each
    input line contains an integer index,
    followed by a tab character, followed by a
    token.

For every input line:
- Output exactly one line.
- Preserve the same INDEX.
- Assign exactly one named entity label.
- Do not skip any indices.
- Do not duplicate any indices.
- Do not renumber the indices.

Output each line in the following format:

INDEX|LABEL

The separator between INDEX and LABEL must be
    the vertical bar character "|".
Do not use tabs.
Do not use spaces.
Correct:
1|B-PER
Incorrect:
1 B-PER
1    B-PER
1: B-PER

The number of output lines must exactly equal
    the number of input lines.

Return only the INDEX|LABEL pairs, one per
    line. Do not include the original tokens,
    explanations, numbering, headings, blank
    lines, or any other text.
"""
```

A.3 Per-Class Inter-Annotator Agreement

Table A1 displays inter-annotator agreement for LID per class. Table A2 displays inter-annotator agreement for NER per class. All disagreements were resolved.

LID Label	Cohen's κ
TR	0.98
EN	0.98
NE	0.94
MIXED	0.75

Table A1: Per-class inter-annotator agreement for language identification (LID).

NER Label	Cohen's κ
B-LOC	1.00
I-LOC	1.00
B-PER	1.00
I-PER	1.00
B-PROD	1.00
B-TITLE	1.00
I-TITLE	1.00
O	0.94
B-OTHER	0.00

Table A2: Per-class inter-annotator agreement for named entity recognition (NER). The low κ score for B-OTHER is likely due to the small number of instances in this category.

A.4 Additional Experimental Results

A.4.1 Per-Class Named Entity Recognition

Table A3 presents per-class named entity recognition results for GPT4o and Qwen.

Table A4 presents per-class named entity recognition for the encoder models.

A.5 Error Analysis Examples

Table A5 presents representative model errors involving Turkish morphological integration and named entity ambiguity.

A.6 Released Resources

The following resources are publicly available:

- TurEngMix Corpus
- TurEngMix Benchmark
- Annotation guidelines

Model	O	ORG	PER	LOC	PROD	GROUP	TITLE	EVENT	TIME	OTHER
GPT-4o Zero-shot	0.98	0.48	0.79	0.83	0.53	0.35	0.69	0.46	0.20	0.29
GPT-4o Few-shot	0.98	**0.57**	**0.83**	**0.84**	**0.67**	**0.42**	**0.72**	**0.43**	0.09	0.34
Qwen3-8B Zero-shot	0.97	0.43	0.55	0.65	0.46	0.35	0.66	0.00	0.09	**0.43**
Qwen3-8B Few-shot	0.98	0.47	0.63	0.64	0.52	0.28	0.53	0.00	0.07	0.34

Table A3: Per-class F1 scores for decoder LLMs on NER task. O denotes non-entity tokens.

Model	O	LOC	ORG	PER	PROD	TITLE	OTHER
Baseline	1.00	0.00	0.00	0.00	0.00	0.00	0.00
XLM-RoBERTa	0.97	0.00	0.00	0.00	0.28	0.00	0.00
BERTurk	0.97	0.00	0.091	0.44	0.27	0.14	0.00
TurkishBERTweet	0.96	0.00	0.00	0.00	0.27	0.06	0.00

Table A4: Per-class F1 scores for named entity recognition using encoder models. O denotes non-entity tokens.

- Dataset construction scripts
- Training and evaluation scripts
- LLM prompting scripts
- Fine-tuning code

Token	Gold Label	Model Prediction	Error Explanation
influencerlar	MIXED	TR	English-origin stem *influencer* with Turkish plural/possessive morphology. The token requires identifying both the English lexical root and Turkish morphological integration.
google'a	B-PROD	B-ORG	In context, *Google* refers to the search product/service rather than the organization.
sony'nin	NE	MIXED	The token contains a Turkish possessive suffix attached to a named entity. The model incorrectly prioritized morphological mixing over named entity identification.

Table A5: Representative error cases involving morphological integration and named entity ambiguity.

Where Does Noise Robustness Live? Evidence from Ablation, Role Swaps, and Cross-Lingual Transfer in Code-Mixed Banking NLU

Piyush Agarwal
Independent Researcher, India
mynameispiyush@gmail.com

Abstract

Conversational banking assistants on messaging platforms face language that their natural-language-understanding (NLU) pipelines were never designed for: vowel-dropped shorthand (*blnc chk krdo*), romanized code-mixing, colloquial verb frames. Layered pipelines stack repair rules, learned retrieval, and an LLM, but which layer carries the noise robustness is rarely measured. We conduct targeted interventions on three candidate robustness-bearing components (the normalizer, rules engine, and conditional LLM arbiter) within a fixed 13-layer Hinglish banking NLU (163 intents), with replacement and transfer probes, on BankStress-330, a released three-tier diagnostic stress set, $\geq$3 repeats per LLM-bearing BankStress-330 configuration: LLM-in-the-loop configurations flip up to $\sim$3% of verdicts across identical temperature-0 runs, defeating single-run ablation. A 2$\times$2 factorial shows the Hinglish normalizer helps a base encoder but is redundant after noisy-query fine-tuning; the post-fine-tuning redundancy also holds on public Hinglish-TOP (n=6,513, McNemar z=5.0, $p\approx 5\times10^{-7}$). Only removing the conditional LLM arbiter collapses adversarial accuracy (.739$\rightarrow$.545); neither tested substitute matched it, yet the same fine-tuned cross-encoder improves both noisy tiers while preserving clean accuracy when restricted to reranking (directional, unresolved at n=330). On native-reviewed Tamil and Telugu probes normalization is unstable in sign. Robustness appears to migrate into tuned components, only for the pair tuned on.

1 Introduction

WhatsApp-register Hindi–English ("Hinglish") banking queries combine four noise sources: orthographic noise (typos, vowel-dropped SMS shorthand), code-mixing (Latin-script Hindi with embedded English banking nouns), colloquial lexical variety (*karza nipta do* for loan foreclosure), and underspecification against a fine-grained intent taxonomy. Pretrained encoders degrade substantially on such input (Kumar et al., 2020; Aliakbarzadeh et al., 2025), and clean or translated test sets overestimate real performance on native noisy queries (He et al., 2026). Production systems answer with layered pipelines: deterministic normalization and rules up front, learned retrieval in the middle, an LLM at the end. Each layer was added for a reason, usually a remembered failure. Whether each layer still *earns its place* is a question deployed systems rarely revisit.

The question is live because the literature answers it both ways. Lexical normalization improved downstream tagging in code-switched settings as recently as 2021 (van der Goot and Çetinoğlu, 2021; van der Goot et al., 2021), yet noise-pretrained or noise-fine-tuned models have repeatedly reduced the value of some input-cleanup methods (Nguyen et al., 2020; Zhuang and Zuccon, 2022; Maarouf and Tanguy, 2025). Contemporary code-mixed fintech assistants, including pipelines with language-classification components, nonetheless still classify on normalized queries (Hazarika et al., 2025).

We revisit the question on a 13-layer NLU built for a WhatsApp banking assistant. The system is a single-team research build; it does not serve live bank customers, and we claim no production deployment. By *production-style* we mean, operationally: enforced latency budgets (median hot-path $\leq$120 ms), an open-weight-only on-premise constraint, a 163-intent taxonomy, and a compliance-driven personally-identifiable-information (PII) layer: constraints that shaped the architecture and that a research prototype would not face. Our method is deliberately simple: remove one layer at a time, substitute candidate replacements, re-measure on a tiered benchmark; then repeat the key configurations on a language pair the pipeline was never tuned for, and repli-

Proceedings of the 11th Workshop on Natural User-generated Text, pages 356–367
October 28, 2026 ©2026 Association for Computational Linguistics

cate the central finding on the public Hinglish-TOP benchmark (n=6,513). We do not claim generality across architectures; we claim the *question* is answerable by measurement, and report the answer for this pipeline. Contributions:

- **C1: Tiered evaluation protocol.** A repeatable measurement protocol (tier-stratified stress queries, 330 over 22 core intents classified against the full 163-intent space; $\geq$3 repeats per LLM-bearing BankStress-330 configuration with variance reported; paired per-query tests), released as the BankStress-330 benchmark with per-query authoring notes and the full label-audit record. The protocol also yields a measured $\sim$3-point single-run verdict noise floor for LLM-in-the-loop ablation. We are not aware of a public Hinglish banking intent benchmark, despite public code-switched and banking benchmarks in adjacent spaces (Khanuja et al., 2020; Aguilar et al., 2020; Casanueva et al., 2020; Agarwal et al., 2023).
- **C2: Factorial layer ablation.** A factorial experiment showing hand-built normalization helps a base encoder but becomes redundant once the encoder is fine-tuned on noisy queries; the rules engine is a latency feature; of the three targeted layers, only the bounded LLM arbiter is load-bearing under the substitutions we test.
- **C3: Replacement and escalation study.** The same fine-tuned cross-encoder fails as a *decider* (.461) and wins as a *ranker* (.906), a mechanism-supported result (shortlist recall, escalation reduction); its end-to-end accuracy delta is directional at our benchmark size. An 8B arbiter recovers only a fraction of the 20B's value; failure-targeted augmentation shows dose-dependent harm. Unlike C2, these findings rest on the in-house benchmark alone (§3). The escalation trigger tracks noise tier without labels (67/73/94%), but its *rescue rate* is inverted ($\sim$26% of adversarial vs. $\sim$13% of clean escalations).
- **C4: Transfer probes.** Across three native-validated unseen-pair sets (two independently native-reviewed Tamil–English renderings; Telugu–English), the normalizer's contribution is unstable in sign (-3.3, -2.5, $+2.9$): off the tuned pair, no layer's role can be assumed.

Read together these are one argument: C1 is the guardrail that makes layer attribution trustworthy; C2 locates robustness in functional roles, not rules or scale; C3 is its sharpest instance; C4 bounds it to the tuned pair.

2 System

The pipeline (Fig. 1) processes each utterance through: (L1) PII masking with checksum validation (Luhn for cards, Verhoeff for national IDs); (L2) a 10-stage Hinglish normalizer ($\sim$80 rule families: vowel-restoration regexes such as *kr*→kar**, Indian numeric grammar such as *lakh/crore* and fraction words *sawa/paune*; with PII spans frozen as placeholders); (L3) language identification (LID; two-stage FastText with a banking-loanword suppression list); (L4) sentence encoding with multilingual-e5-base plus a low-rank-adaptation (LoRA) adapter fine-tuned on 4,300 hard-negative triplets and 1,600 noisy queries; (L5) centroid-based domain routing; (L6) a deterministic rules engine (fast exit at confidence $\geq$0.95); (L7) per-domain hybrid retrieval (dense FAISS nearest-neighbor search + lexical BM25, reciprocal-rank fusion, lexical weight downscaled for stopword-heavy queries); (L8) score-cliff detection; (L9) a 7-signal out-of-scope ensemble (cf. Larson et al., 2019); (L10) a decision matrix emitting EXECUTE/RERANK/CLARIFY/OOS; (L11) a *conditional* LLM arbiter (gpt-oss-20b, an open-weight model prompted to pick among retrieved candidates), invoked only when the decision matrix cannot certify a clear winner; (L12) risk-aware confidence adjustment; (L13) slot extraction.

Two design properties matter for what follows. The pipeline *abstains into arbitration*: the escalation decision is deterministic and precedes any LLM call. And the arbiter is bounded: it selects among retrieved candidates rather than free-generating, and its output passes through downstream confidence gates.

3 Experimental setup

Benchmark. We construct and release **BankStress-330**, a diagnostic stress set for layer attribution (not a leaderboard benchmark): 330 queries over 22 core banking intents: CLEAN (66; canonical phrasings), MESSY (154; typos,

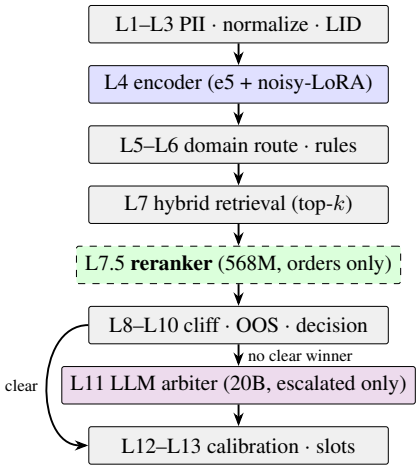


Figure 1: The 13-layer pipeline (condensed). Gray layers are deterministic; blue is the learned encoder; green (dashed) is the reranker intervention evaluated in §4.4; violet is the conditional LLM arbiter.

shorthand, code-mix), ADVERSARIAL (110; colloquial idioms, implicature, zero domain keywords). Each query is classified against the full 163-intent space. Because the benchmark was authored by the same author who built the system, its notion of "hard" co-evolved with the pipeline; of our four findings, only the normalizer result is additionally replicated on public data; the arbiter and reranker findings rest on this benchmark alone and are scoped accordingly; conversely, every structural claim with a public vehicle is checked on public data (§4.1). The set is deliberately confusion-heavy; escalation rates on it are stress-set figures (production-representative traffic, i.e., internally generated test conversations replaying realistic usage, no real customers, escalates $\sim$5% of turns, §6), not traffic estimates.

Annotation provenance. BankStress-330 was authored and initially labeled by the sole author. The same author later audited the 38 majority-vote failure labels with system predictions visible (§5); 35 labels were retained and 3 corrected. This is a label audit, not independent double annotation: no inter-annotator agreement is claimed for BankStress-330, and the full audit record is released so it can be repeated independently. Each Tamil/Telugu transfer set was reviewed by one native-speaking volunteer.

Protocol. All LLM-bearing BankStress-330 configurations were evaluated with at least three repeats; the public Hinglish-TOP arbiter comparison is a single-run exploratory control (§4.3). Deterministic configurations are reported once after verifying bit-exact outputs. All runs use one machine (CPU inference for the pipeline; the LLM arbiter served via API at temperature 0; all models in the stack are open-weight and on-premise deployable, a banking compliance requirement). We report per-tier accuracy as mean with min–max across repeats, exact McNemar on majority-vote failure sets, and paired bootstrap CIs over per-query correctness. This is a targeted, non-exhaustive ablation study: we intervene on three candidate robustness-bearing layers (the normalizer, rules engine, and conditional LLM arbiter); the other ten layers are held fixed throughout. Ablations stub exactly one layer; the reranker experiment inserts exactly one component between L7 and L8.

Reproducibility observation. Configurations without the LLM arbiter reproduce *bit-identically* across repeats run under identical harness code;[1] configurations containing it flip 1–3% of per-query verdicts between identical temperature-0 runs (3–9 per 330 across all 21 pairings of 7 released baseline repeats, mean $\sim$2%; 15/330 queries flip in at least one repeat). The deterministic layers are exactly reproducible; the LLM contributes all run-to-run variance. This has a methodological implication beyond our study: any *single-run* ablation of an LLM-in-the-loop pipeline carries a $\sim$3-point verdict noise floor, so differences below that require repeated-measures testing, which is why every BankStress-330 comparison in this paper uses $\geq$3 repeats for LLM-bearing configurations, while deterministic configurations (including the public-data normalization runs) are single-run bit-exact under fixed code and weights (the public Hinglish-TOP arbiter comparison, which invokes the LLM, is a single-run exploratory control); we report three decimals for traceability but do not interpret differences within this noise floor. A run ledger maps every table to its result files, run counts, and harness era (`MANIFEST.md` in the artifacts). We do not correct for multiple comparisons across the many configurations tested: the load-bearing contrasts survive any standard correction by orders of magnitude, and the borderline ones (reranker,

[1] A mid-study harness patch (adding per-query logging and ablation stubs) shifted exactly one –arbiter query (.761→.758) and cut deterministic –arbiter p50 from $\sim$60 to $\sim$22 ms; both eras' files are released and labeled, and the Table latency figures are from the pre-patch era.

Config	CLN	MSY	ADV	Overall
Full pipeline	.985	.948	.739	.886
– Normalizer	.980	.961	.752	.895
– Rules engine	.975	.951	.739	.885
– LLM arbiter	.924	.844	**.545**	.761

Table 1: Layer ablation (mean of 3 repeats; ranges in text); higher-without-layer rows mean the layer was net-neutral or harmful post-fine-tuning. Median (p50) latency: ∼440 ms with the arbiter, ∼60 ms without.

Encoder	Norm. ON	Norm. OFF	Norm. effect
Base	.857	.833	+2.4 (+5.5 **adv**)
Noisy-FT	.886	.895	−0.9 (n.s.)

Table 2: Factorial: encoder training × normalizer (overall accuracy; 3 repeats/cell). The normalizer helps the base encoder and is redundant after noisy fine-tuning.

transfer) are interpreted as directional precisely because they would not.

4 Results

4.1 Which layers carry noise robustness

On our benchmark, removing the hand-built normalizer does not reduce accuracy (Table 1; 7 fixed vs. 7 broken discordant queries on majority-vote failure sets over the released repeats; p=1.0): the rules occasionally rewrite away signal: removing the normalizer *fixes* a typo query (*mera **crd** card chori ho gya*). The rules engine is a latency feature: accuracy is unchanged without it. Only the arbiter's removal collapses the adversarial tier (−19.4 points; 51 queries saved vs. 6 cost on majority-vote failure sets over the released repeats; p=5.7×10^{-10}).

A factorial experiment isolates the cause (Table 2). We rebuilt the pipeline's indices with the *base* encoder (no noisy-query LoRA) and crossed encoder training with the normalizer. Before noisy fine-tuning the normalizer genuinely helps, consistent with positive extrinsic results in the normalization literature (van der Goot and Çetinoğlu, 2021; van der Goot et al., 2021). After fine-tuning on 1,600 noisy queries its benefit disappears. The interaction supports the reading that noisy fine-tuning made normalization redundant in the tested settings (cf. Nguyen et al., 2020; Zhuang and Zuccon, 2022). The fine-tune itself is worth +2.9 to +6.2 points depending on the normalizer condition. **The post-fine-tuning redundancy replicates on public data** (the base-encoder benefit does not: on Hinglish-TOP the normalizer has no effect even untuned, consistent with domain-matched rules being its precondition). We rebuilt the pipeline's taxonomy and indices from the Hinglish-TOP training split (Agarwal et al., 2023) (58 intents; reminders, weather, media; no banking) and evaluated its 6,513-query human-annotated test set (105 duplicate surface forms in the public split are retained as distinct examples) in the deterministic no-arbiter configuration (single runs are bit-exact). With the noisy-query LoRA, removing the normalizer *improves* accuracy .617→.631 (fixes 202 queries, breaks 112; McNemar z=5.0, $p \ll 0.001$); with the base encoder the normalizer has no effect (≈.603 both: .6028 vs .6033). Two refinements follow: the encoder's noise robustness transfers cross-domain (+1.4 to +2.7 over base), and the normalizer only ever helped when the encoder was untuned *and* its rules were domain-matched (our in-house base-encoder cell). We use *migrate* operationally: a layer's accuracy contribution becomes indistinguishable from zero while the encoder's grows, and reappears on a pair the encoder was not tuned for (§4.5).

4.2 Escalation rises with difficulty; rescue efficiency stays low

The decision matrix escalates 67% of CLEAN, 73% of MESSY, and 94% of ADVERSARIAL queries; escalation rate tracks tier difficulty with no difficulty labels anywhere in the system. But comparing baseline against the –arbiter configuration reveals the arbiter's *rescue rate* (measured indirectly: the per-query outcome delta between the two configurations under a rest-of-pipeline invariance assumption): the arbiter turns a wrong answer into a correct one for ∼26% of escalated adversarial queries (24.3–27.2 across 21 pairings of independent baseline and –arbiter repeats), ∼16% of messy (15.9 across all pairings), and only ∼13% of clean ones (11.4–13.6); the reverse (harm) direction is what the paired repeats' spread reflects, and on public data the two directions cancel (§4.3). Escalation *rates* carry no such interval: the trigger is upstream of the LLM and reproduces exactly. Most clean escalations are risk-averse confirmation (confusion-group policy), not necessity: 44 clean-tier LLM calls yield ∼6 rescues against ∼1 harm; in-domain harm stays at a mean of 1–4% of escalations on every tier (0–5% across individual pairings), whereas off-domain rescue and harm

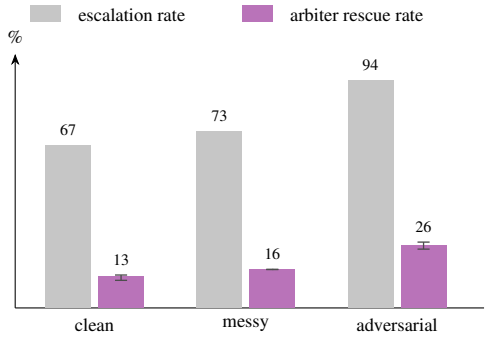


Figure 2: Escalation rate tracks noise tier, but the arbiter's *rescue rate* (share of escalations where it turns a wrong answer into a correct one) is lowest exactly where escalation is cheapest to skip. Whiskers: min–max across 21 pairings of independent baseline and –arbiter repeats; rates are deterministic and carry no interval.

Attempt	Overall	ADV
Arbiter substitutes		
Zero-shot xenc as decider	.530	.336
Fine-tuned xenc as decider	.461	.282
8B instruct LLM as arbiter	.810	.579
Augmentation (contrast)		
Full dose	.873	.758
Half dose (holdout eval)	.893	.757

Table 3: Interventions that did not pay off. Rows 1–3 substitute the arbiter; rows 4–5 are augmentation (a different intervention, shown for contrast). Note the half-dose row exceeds baseline overall yet fixes only 2/21 held-out failures: targeted gains, no generalization.

cancel (§4.3). Per-query latency splits at ∼70 ms (hot path in the baseline configuration; Table 1's ∼60 ms is the –arbiter configuration's overall p50) vs. 417–530 ms (escalated). Figure 2 visualizes the inversion.

The wasted-escalation problem has begun to be named in the cascade literature (Jitkrittum et al., 2023; Rabanser et al., 2025; Bouchard, 2026); stratifying escalation and rescue rates by input noise tier is, to our knowledge, new, and it identifies exactly where a cheaper intermediate stage would pay, which §4.4 exploits.

4.3 What does not replace the arbiter

A 568M multilingual cross-encoder (bge-reranker-v2-m3, a reranking fine-tune released alongside M3-Embedding; Chen et al., 2024; Beijing Academy of Artificial Intelligence (BAAI), 2024) substituted for the arbiter fails zero-shot (colloquial-Hinglish-to-intent matching is out-of-distribution for passage-relevance training) and fails *worse* after fine-tuning on 21k taxonomy pairs: it learns to recognize familiar phrasings, but escalated queries are precisely the unfamiliar ones (Table 3). Rerankers also emit relative orderings, not calibrated confidences; forcing their scores through absolute-confidence gates misuses them (Yan et al., 2022). A smaller *decoder* fails too: swapping the 20B arbiter (gpt-oss-20b) for an 8B instruct model (Llama-3.1-8B-Instruct; same prompt, same role; 3 repeats) recovers only a fraction of its value: adversarial .573–.591 across repeats (the table reports the mean), barely above running *no* arbiter (.545) and far below the 20B's .739. We report this as a model-swap observation (none .545 → 8B ∼.58 → 20B .739): the two arbiters differ in training data and instruction tuning, not only in size, so we cannot attribute the gap to parameter count alone. The arbiter's value is also domain-bound: on public Hinglish-TOP (non-banking, 58 intents), enabling it over the deterministic configuration is a single-run exploratory control, not a repeated comparison; in this single recorded public-data run the arbiter produced no detected gain (.6323 vs. .6306, +0.17pp, n=6,513; 959 queries fixed vs. 948 broken, exact McNemar p=0.82; 76% invocation). In this run it recovers 39.9% of the deterministic failures while breaking a nearly equal number of successes (959 fixed vs. 948 broken), so its large in-domain value (Table 1) does not transfer off the banking distribution; run files are released.

The augmentation rows document a known risk (He et al., 2023; Joshi and He, 2022) materializing, with a dose–response. Generating colloquial phrases from observed failures (practice: Wei and Zou, 2019; Sahu et al., 2022, 2023) at full dose (1,303 accepted phrases) shifts intent centroids enough to break the easy head (CLEAN −4.1); accumulation alongside real data, the standard remedy in the synthetic-data literature (Gerstgrasser et al., 2024), did not prevent it. At half dose the head survives intact, but only 2 of 21 *held-out* failures are fixed. Failure-targeted augmentation fixes what it targets, barely generalizes to untargeted failures, and harms broadly only as the dose grows (cf. Li et al., 2023; Ma and Zhang, 2026).

4.4 What helps: the same model, in its proper role

Inserting the *fine-tuned* cross-encoder between retrieval and score-cliff (rescoring the top-8, blending with retrieval scores, reordering, deciding

Config	MSY	ADV	Overall	Esc.
Baseline	.948	.739	.886	260
+ zero-shot rerank	.929	.779	.890	210
+ **FT rerank**	**.961**	**.782**	**.906**	242

Table 4: Reorder-only reranking (α=0.3 blend; mean of three designated table runs, see `MANIFEST.md`; CLEAN is .985 throughout). Esc. = LLM escalations per 330. The fine-tuned reranker improves both noisy tiers' means while preserving clean accuracy; the zero-shot variant trades MESSY for ADVERSARIAL.

Config	Overall	p50 (ms)	LLM calls
– LLM arbiter	.761	∼60	0
8B arbiter	.810	∼405	260
Baseline (20B)	.886	∼435	260
+ zero-shot rerank	.890	∼425	210
+ FT rerank	**.906**	∼460	**242**

Table 5: Accuracy–cost frontier across configurations (stress set).

nothing) measurably improves the candidate list the arbiter sees. The harness logged each query after normalization; joining each of the 308 reranked entries to its benchmark record through a checked, order-preserving mapping (released with the artifact so every association is inspectable, rather than the biased exact-string join that retains only the queries normalization left byte-identical and no clean ones), shortlist recall@3 rises from .864 to .886 (+2.3 points) across all reranked queries and is positive under every recovery method we tried. Recall@1 decreases from .740 to .708 (−3.2 points), so the reranker improves shortlist coverage at k=3 while worsening top-1 retrieval: it reshapes the depth-3 envelope the arbiter selects from rather than the top-1 prediction, consistent with the same weights scoring only .461 as a standalone decider. In parallel, adversarial escalations fall from 94% to 85% while adversarial accuracy *rises* (Table 4). The same weights yield an end-to-end pipeline at .461 when substituted as the decider and at .906 when restricted to ranking. Statistically: across every released repeat of both arms (15 reranked, 16 baseline; all runs reported, none dropped), reranked runs match or exceed baseline runs in 231 of 240 pairwise comparisons; the 9 exceptions are the three *zero-shot* bge-reranker runs paired with three baseline repeats, and restricting to the 12 fine-tuned reranker runs gives 192 of 192. Pooling the zero-shot and fine-tuned arms, which Table 4 reports separately, into one reranked set is the conservative choice. The paired per-query bootstrap gives +1.6 overall with 95% CI $[0.0, +3.3]$; the lower bound sits at zero. The coarser majority-vote McNemar is non-significant (p=0.18). We therefore claim a consistent directional gain that n=330 cannot resolve to significance, and rest the finding on the mechanism measurements rather than on the accuracy delta. Hyperparameter provenance: the blend weight was chosen from a two-value probe ($\alpha \in \{0.5, 0.3\}$, single runs) on this same benchmark, and the shortlist depth is the retriever's fixed top-8; no disjoint tuning split existed at this scale, so the +1.6 directional estimate should additionally be read as optimistic. In this cross-encoder experiment, role restriction, not model scale, determined whether it helped or harmed the pipeline, consistent with ranking theory: learning-to-rank objectives leave scores scale-uncalibrated, so relative order is trustworthy where absolute thresholds are not (Yan et al., 2022; Yu et al., 2025; Baral et al., 2026). Prior pairwise intent deciders succeeded in *clean few-shot* settings (Zhang et al., 2020); in our high-label-space, colloquial, escalated-query regime, the decider role fails while shortlist reranking succeeds. The closest system-level prior routes between two deciders on uncertainty (Arora et al., 2024); restructuring the arbiter's candidate list is a different, and here better, use of the small model. Table 5 summarizes the accuracy–cost frontier: the fine-tuned reranker improves accuracy while reducing LLM calls. Its measured p50 does not establish an end-to-end speedup: the reranker adds local compute, LLM-bearing p50 is API-dominated and varied across runs, and the FT-rerank p50 is if anything higher than baseline. The meaningful cost contrast is LLM calls, not latency. The no-arbiter configuration is a separate frontier point, viable wherever 12 points of accuracy can be traded for the removed API round-trip.

4.5 Transfer: off-distribution, layer roles become unstable

Romanized Dravidian–English code-mix has its own literature and shared tasks (Chakravarthi et al., 2022); intent-labeled resources, however, remain scarce. We had a native Tamil speaker review and correct machine-generated candidate renderings of a stratified 80-query slice (20/30/30 across tiers) in Tanglish, romanized Tamil–English code-mix as a Tamil customer would type it. No

	ta-T1 (80)	ta-T2 genz (80)	te (40)
Baseline	.800	.825	.742
– Normalizer	.767 ↓	.854 ↑	.717 ↓
– LLM arbiter	.762	–	.625
+ FT reranker	.788 ↓	–	.708 ↓

Table 6: Transfer probes: two independently native-reviewed Tamil–English renderings (standard register T1; Gen-Z register T2) and Telugu–English, all native-reviewed, zero-shot with respect to this study (no Tamil or Telugu task-specific training or adaptation; the underlying multilingual encoder may have encountered these languages during pretraining). Arrows: direction of change vs. each set's own baseline. Empty cells: configurations not run on T2, where the compute budget prioritized the normalizer contrast; the reranker and arbiter transfer claims therefore rest on T1 and Telugu only.

Tamil or Telugu task-specific training or adaptation was performed in this study: the normalizer is Hinglish-tuned, the LoRA saw only Hinglish noise, and the reranker was fine-tuned on the Hinglish taxonomy; the underlying multilingual encoder may have encountered these languages during pretraining. Four observations (Table 6). First, zero-shot transfer is viable on every set (.74–.83): multilingual encoding plus English banking loanwords carry most of the signal. Second, and centrally: **the normalizer's contribution off-distribution is unstable in sign**. It helps on standard-register Tanglish (-3.3 when removed; paired bootstrap 95% CI $[-8.1, +0.9]$; all three –normalizer repeats fall below all three baseline repeats, but, like the reranker effect, the effect is not resolved to significance at n=80) and on Telugu (-2.5 in the mean; the Telugu repeats overlap the baseline range, so this set supports mean-consistency only), but, as observation three shows, not universally. A stage-wise ablation on the Tanglish set attributes $+2.1$ of the benefit to unambiguously language-agnostic stages and $+1.2$ to the strictly Hindi-specific stages, jointly accounting for the full 3.3-point gap. Third, **the direction reverses on the Gen-Z-register Tamil set** ($+2.9$ when removed, with all three –normalizer repeats *above* all three baseline repeats; the T1/T2 sign contrast, opposite and repeat-consistent on both sides, is the robust observation): the normalizer's sign is unstable across two independent renderings of the same content. Because each register has a different native reviewer, register and reviewer idiosyncrasy are confounded (Limitations) and we do not attribute the reversal to register alone; either reading warns that single-reviewer transfer evaluations can mislead in both direction and magnitude. Fourth, two effects are stable across every set where tested: the Hinglish-fine-tuned reranker improves neither tested pair and the LLM arbiter remains load-bearing (Telugu: -11.7 without it); both were tested on T1 and Telugu only, not on T2. We state the scope plainly: n=40–80 per set, one native reviewer per set, a loanword-heavy domain that likely flatters transfer. The safe architecture lesson is accordingly conservative: off-distribution, no layer's contribution can be assumed: it must be measured per pair and register. Zero-shot transfer stayed viable everywhere (.74–.83), consistent with the encoder carrying the transferable signal; every other layer was pair-dependent, and even the arbiter is a wash on public non-banking data (§4.3).

5 What remains broken

If robustness now lives in learned components and a bounded arbiter, what still defeats all of them? The residual-failure taxonomy uses majority voting (fail in $\geq$ half the runs) over four baseline repeats (identified in the artifact manifest as the set whose majority-failure set reproduces the label-audit record): 38 majority failures, with 43 failing in at least one repeat (across all seven released repeats the counts are 35 and 46; we use the four-repeat set because its majority-failure set matches the label-audit record). These decompose into: **(F1) colloquial lexical variety** ($\sim$16; *kitna bacha mera*, *4 digit wala code bhool gya*; nothing misspelled, so no normalizer can help; retrieval lacks the verb frames); **(F2) pragmatic implicature** ($\sim$6; *credit card chhota pad rha h* = limit increase; the arbiter's remaining ceiling); **(F3) sibling-intent granularity** ($\sim$14; statement↔balance↔outstanding; a human would ask a clarifying question; forcing argmax is the error); **(F4) multi-question utterances**; **(F5) context-dependent fragments** (*baaki kya hai*), strictly unanswerable in a stateless evaluation. These are largely not addressable by orthographic normalization alone. An oracle analysis quantifies F3: 12/38 failures are wrong-intent-but-same-domain; converting them to clarification questions lifts the ceiling from .886 to .921, and only 5 of the 12 fall inside the system's *declared* confusion groups, i.e., the hand-maintained am-

biguity configuration under-covers real ambiguity by more than half. All 38 majority-vote failure labels were audited by the benchmark author: 35 were retained and 3 corrected (e.g., *Show my debit card details*, gold *dc_explore*, corrected to *dc_card_details*). System predictions were visible during the audit, which risks anchoring; we release the full record so the audit can be independently repeated. The audit shifts headline accuracy by about $+0.7$ points on average (baseline .886→.893; individual runs move by at most $+0.9$) and changes no conclusion; we report raw numbers throughout and release the label-audit record with the benchmark.

6 Evolution: measurement-driven design

The pipeline reached its current shape through five measurement→defense iterations; e.g., romanized-Hinglish LID accuracy collapsed to 6.3% under banking loanwords in a development-time measurement (whose logs predate this study and are not part of the release) before a loanword-suppression list fixed it, and LLM invocation fell from 20–30% of turns to a conditional $\sim$5% on production-representative traffic (the 330-query stress set is deliberately confusion-heavy, hence its 67–94% escalation rates in §4.2). This paper is the sixth iteration of that loop, pointed at the pipeline itself.

7 Discussion

Where robustness lives. Not in the rules: three generations of repair-rule writing were made redundant *for the tuned pair* by one fine-tune on noisy queries (§4.1). It lives in (a) the noisy-text encoder, (b) a task-fine-tuned ranker over its candidates, (c) a bounded LLM arbiter for the residue; and, for pairs outside the fine-tuning distribution, (d) a mixture re-measured per pair and register (our scaffolding helped on two unseen sets and hurt on a third). The residual gap is lexical variety and taxonomy granularity, arguing for clarification actions and colloquial coverage, not more repair rules. The arbiter was API-served, though every model in the stack is open-weight and on-premise deployable; the full-dose augmentation result is in-loop by construction (the half-dose hold-out addresses this); stress-set escalation rates do not reflect production traffic.

Strength of evidence. Our findings differ in evidential weight. The normalizer finding rests on BankStress-330 plus the public Hinglish-TOP replication. The arbiter finding is supported on BankStress-330 alone. The reranker finding is BankStress-only, and its end-to-end accuracy improvement is directional, not statistically resolved. The Tamil and Telugu transfer findings are small, single-reviewer directional probes. Generalization beyond banking and beyond this pipeline is not established.

8 Related work

Lexical normalization. The W-NUT Multi-LexNorm shared task established multilingual lexical normalization and reported *positive* extrinsic effects on parsing and tagging (van der Goot et al., 2021; Samuel and Straka, 2021; van der Goot and Çetinoğlu, 2021). Our results do not contradict these findings; they date them. With encoders pretrained or fine-tuned on noisy text the picture inverts: dictionary normalization stops helping a tweet-pretrained LM (Nguyen et al., 2020), LLM normalization slightly *hurts* sequence labeling on noisy text (Maarouf and Tanguy, 2025), and typo-robust *training* outperforms a rule-based correction baseline in dense retrieval in that study (Zhuang and Zuccon, 2022). We re-run the question on a production code-mixed intent pipeline and find the 2021 conclusion no longer holds (§4.1), except on pairs outside the fine-tuning distribution (§4.5); a contemporary Hinglish fintech assistant, whose pipeline also includes language classification, still classifies intents on normalized queries (Hazarika et al., 2025).

Code-mixed NLU. Hindi–English code-switching benchmarks contain no intent-classification task (Khanuja et al., 2020; Aguilar et al., 2020; Sheth et al., 2025); Hinglish-TOP covers task-oriented parsing in non-banking domains (Agarwal et al., 2023), BANKING77 is English-only (Casanueva et al., 2020), and MASSIVE is translated rather than code-mixed (FitzGerald et al., 2023). Realistic-noise studies report clean→noisy gaps of 2–4 points (Aliakbarzadeh et al., 2025); our messy tier matches this, while our adversarial tier (-24 points) shows colloquial code-mix far exceeds "typo noise". We refer to Winata et al. (2023) for a broad survey of code-switching NLP.

Cascades and routing. LLM cascades and routers reduce cost by sending easy inputs to

cheap models (Chen et al., 2023; Ding et al., 2024; Ong et al., 2025; Varangot-Reille et al., 2025), and can improve accuracy, not just cost (Varshney and Baral, 2022). Recent work begins to name the wasted-escalation problem (Jitkrittum et al., 2023; Rabanser et al., 2025; Bouchard, 2026; Aggarwal et al., 2023); we add a measurement it lacks: escalation and rescue rates stratified by input noise tier. The closest system routes intent queries between a classifier and an LLM on uncertainty (Arora et al., 2024), choosing *between two deciders*, whereas we show restructuring the LLM's candidate list with a non-deciding ranker beats replacing either component (§4.4 explains this via ranking theory).

Augmentation pitfalls. Text augmentation is widely used (Wei and Zou, 2019), including LLM-based variants for intent classification (Sahu et al., 2022, 2023), with known label-noise risks near class boundaries. Prior work anticipates that failure-directed augmentation harms other subgroups (He et al., 2023), counterfactual augmentation suppresses robust features (Joshi and He, 2022), and theory predicts synthetic data hurts when class imbalance is not the dominant error source (Ma and Zhang, 2026). Gerstgrasser et al. (2024) show that accumulating synthetic data alongside real data averts *recursive* model collapse; that setting differs from our one-stage augmentation, where accumulation alongside the real training data did not prevent the harm we document. Robustness benchmarks for task-oriented dialogue (Liu et al., 2021; Peng et al., 2021) perturb inputs but do not ablate deployed pipelines layer-by-layer.

9 Conclusion

We asked where noise robustness lives in one production-style code-mixed banking NLU, targeting three candidate layers within the fixed 13-layer pipeline, and find it migrates: of the three targeted layers, repair rules carried it before encoder fine-tuning, afterward the encoder does, but only for the pair it saw. Deep-noise decisions stay with a bounded LLM arbiter no substitute matched; that same model, restricted to reranking, lifted recall and cut escalations. Layers in mature pipelines have measurable *half-lives*; factorial ablation crossed with training condition, plus a transfer probe, separates load-bearing defenses from vestigial ones. The protocol is system-agnostic.

Limitations

Single domain (banking), whose heavy English loanwords likely flatter zero-shot transfer. Four evaluation sets across three code-mix pairs (Hinglish; two independently native-reviewed Tamil–English renderings, one per register, so reviewer identity and register are confounded; Telugu–English at n=40); each transfer set is single-reviewer and small. The in-house benchmark has author-assigned, author-audited labels with no independent double annotation (the normalizer finding is additionally replicated on public Hinglish-TOP). The reranker's end-to-end accuracy effect (+1.6 points) is not resolved to significance at n=330 by any test we report (bootstrap CI lower bound at zero; McNemar p=0.18); the mechanism measurements (shortlist recall and escalation reduction) are its firmest support. The arbiter rescue rate (§4.2) is measured indirectly, as the outcome delta between the baseline and –arbiter configurations, under a rest-of-pipeline invariance assumption. The arbiter runs a single prompt at temperature 0; we did not quantify prompt sensitivity, so part of its measured value may be prompt-specific. The arbiter and reranker findings rest on the self-authored benchmark alone; only the normalizer finding has a public-data replication. Finally, we do not evaluate a direct standalone prompted SLM/LLM classifier over all 330 BankStress items; the 8B experiment is a component substitution inside the pipeline and should not be interpreted as such a baseline.

Ethics Statement

All benchmark queries were authored by the author in the style of WhatsApp banking messages; none derive from real customer communications, and the customer-data platform used by the system is entirely synthetic (fictional profiles), so no personally identifiable information of any real individual appears in the system, the benchmark, or this paper. The Tamil and Telugu transfer sets were produced by consenting adult native-speaker collaborators (personal acquaintances of the author) who volunteered without monetary compensation. The task was bounded (reviewing or correcting two machine-generated candidate renderings per query; the released sets contain 80, 80, and 40 reviewed query rows respectively, one reviewer per set, a few hours of work in total), participation

was declinable at any point, and no from-scratch translation was required; participation was voluntary and the collaborators are thanked in the Acknowledgments. Each transfer set was reviewed by one native-speaking volunteer. We acknowledge community norms favoring paid annotation and scope these sets accordingly as small validation probes rather than crowd-sourced resources. LLM API calls made during evaluation contained only synthetic queries. Misrouting a banking intent has real cost (executing the wrong consequential action), so the system requires explicit user confirmation before any high-risk action and abstains (CLARIFY/OOS) rather than guessing under low confidence; dialect and language coverage is uneven and untested pairs may be served worse, which is why we scope claims to the tuned pair. The system is a research build and serves no live banking customers; we make no deployment claims. An LLM-based coding and writing assistant was used throughout this work (for experiment tooling, analysis, and manuscript drafting) under author direction. The BankStress-330 queries are human-authored; the transfer-set candidate translations were machine-generated (llama-3.3-70b) and no machine-generated translation or label was accepted without native-speaker review; benchmark labels were assigned and subsequently audited by the author. All experimental designs, data, and claims were specified and verified by the author, who takes full responsibility for the content.

Acknowledgments

We thank the native-speaker volunteers who reviewed the Tamil and Telugu transfer sets, and the W-NUT 2026 reviewers and organizers for their feedback.

References

Anmol Agarwal, Jigar Gupta, Rahul Goel, Shyam Upadhyay, Pankaj Joshi, and Rengarajan Aravamudhan. 2023. CST5: Data augmentation for code-switched semantic parsing. In *Proceedings of the 1st Workshop on Taming Large Language Models: Controllability in the era of Interactive Assistants!*

Pranjal Aggarwal, Aman Madaan, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, Shyam Upadhyay, Manaal Faruqui, and Mausam. 2023. AutoMix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of LREC*.

Amirhossein Aliakbarzadeh, Lucie Flek, and Akbar Karimi. 2025. Exploring robustness of multilingual LLMs on real-world noisy data. *arXiv preprint arXiv:2501.08322*.

Gaurav Arora, Shreya Jain, and Srujana Merugu. 2024. Intent detection in the age of LLMs. In *Proceedings of EMNLP: Industry Track*.

Aditeya Baral, Radoslav Ralev, Iliya Sotirov Zhechev, Srijith Rajamohan, and Jen Agarwal. 2026. Closing the operational gap in semantic caching. *arXiv preprint arXiv:2606.19719*.

Beijing Academy of Artificial Intelligence (BAAI). 2024. bge-reranker-v2-m3. HuggingFace model card. XLM-RoBERTa-large-based multilingual reranker, 568M parameters. Accessed 2026-09-05.

Dylan Bouchard. 2026. Is escalation worth it? a decision-theoretic characterization of LLM cascades. *arXiv preprint arXiv:2605.06350*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for Conversational AI*.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2022. DravidianCodeMix: Sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text. *Language Resources and Evaluation*. ArXiv:2106.09460.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid LLM: Cost-efficient and quality-aware query routing. In *Proceedings of ICLR*.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan,

Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1M-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. 2024. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. *arXiv preprint arXiv:2404.01413*.

Bharatdeep Hazarika, Arya Suneesh, Prasanna Devadiga, Pawan Kumar Rajpoot, Anshuman B Suresh, and Ahmed Ifthaquar Hussain. 2025. Multilingual conversational AI for financial assistance: Bridging language barriers in Indian FinTech. *arXiv preprint arXiv:2512.01439*.

Haoyu He, Jinyu Zhuang, Haoran Chu, Shuhang Yu, Hao Wang, and Kunpeng Han. 2026. From synthetic to native: Benchmarking multilingual intent classification in logistics customer service. *arXiv preprint arXiv:2603.23172*.

Zexue He, Marco Tulio Ribeiro, and Fereshte Khani. 2023. Targeted data generation: Finding and fixing model weaknesses. In *Proceedings of ACL*.

Wittawat Jitkrittum, Neha Gupta, Aditya Krishna Menon, Harikrishna Narasimhan, Ankit Singh Rawat, and Sanjiv Kumar. 2023. When does confidence-based cascade deferral suffice? In *Advances in Neural Information Processing Systems*.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of ACL*.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of ACL*.

Ankit Kumar, Piyush Makhija, and Anuj Gupta. 2020. Noisy text data: Achilles' heel of BERT. In *Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT)*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. In *Proceedings of EMNLP*.

Jiexi Liu, Ryuichi Takanobu, Jiaxin Wen, Dazhen Wan, Hongguang Li, Weiran Nie, Cheng Li, Wei Peng, and Minlie Huang. 2021. Robustness testing of language understanding in task-oriented dialog. In *Proceedings of ACL*.

Zhengchi Ma and Anru R. Zhang. 2026. Synthetic augmentation in imbalanced learning: When it helps, when it hurts, and how much to add. *arXiv preprint arXiv:2601.16120*.

Mariame Maarouf and Ludovic Tanguy. 2025. Automatic normalization of noisy technical reports with an LLM: What effects on a downstream task? In *Proceedings of the Tenth Workshop on Noisy and User-generated Text (W-NUT)*.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of EMNLP: System Demonstrations*.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. RouteLLM: Learning to route LLMs with preference data. In *International Conference on Learning Representations (ICLR)*.

Baolin Peng, Chunyuan Li, Zhu Zhang, Chenguang Zhu, Jinchao Li, and Jianfeng Gao. 2021. RADDLE: An evaluation benchmark and analysis platform for robust task-oriented dialog systems. In *Proceedings of ACL-IJCNLP*.

Stephan Rabanser, Nathalie Rauschmayr, Achin Kulshrestha, Petra Poklukar, Wittawat Jitkrittum, Sean Augenstein, Congchao Wang, and Federico Tombari. 2025. Gatekeeper: Improving model cascades through confidence tuning. *arXiv preprint arXiv:2502.19335*.

Gaurav Sahu, Pau Rodriguez, Issam Laradji, Parmida Atighehchian, David Vazquez, and Dzmitry Bahdanau. 2022. Data augmentation for intent classification with off-the-shelf large language models. In *Proceedings of the 4th Workshop on NLP for Conversational AI*.

Gaurav Sahu, Olga Vechtomova, Dzmitry Bahdanau, and Issam Laradji. 2023. PromptMix: A class boundary augmentation method for large language model distillation. In *Proceedings of EMNLP*.

David Samuel and Milan Straka. 2021. Úfal at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT)*.

Rajvee Sheth, Himanshu Beniwal, and Mayank Singh. 2025. COMI-LINGUA: Expert annotated large-scale dataset for multitask NLP in Hindi-English code-mixing. In *Findings of the Association for Computational Linguistics: EMNLP 2025*.

Rob van der Goot and Özlem Çetinoğlu. 2021. Lexical normalization for code-switched data and its effect on POS tagging. In *Proceedings of EACL*.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešić, Özlem Çetinoğlu, Rahmad Mahendra, Talha Çolakoğlu, Timothy Baldwin, Tommaso Caselli, and Wladimir Sidorenko. 2021. Multilexnorm: A shared task on multilingual lexical normalization. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT)*.

Clovis Varangot-Reille, Christophe Bouvard, Antoine Gourru, Mathieu Ciancone, Marion Schaeffer, and François Jacquenet. 2025. Doing more with less: A survey on routing strategies for resource optimisation in LLM-based systems. *arXiv preprint arXiv:2502.00409*.

Neeraj Varshney and Chitta Baral. 2022. Model cascading: Towards jointly improving efficiency and accuracy of NLP systems. In *Proceedings of EMNLP*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of EMNLP-IJCNLP*.

Genta Indra Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Findings of ACL*.

Le Yan, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Scale calibration of deep ranking models. In *Proceedings of KDD*.

Puxuan Yu, Daniel Cohen, Hemank Lamba, Joel R. Tetreault, and Alejandro Jaimes. 2025. Explain then rank: Scale calibration of neural rankers using natural language explanations from LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22716–22730.

Jianguo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of EMNLP*.

Shengyao Zhuang and Guido Zuccon. 2022. CharacterBERT and self-teaching for improving the robustness of dense retrievers on queries with typos. In *Proceedings of SIGIR*.

A Artifact Availability

We release, under CC BY 4.0 (data) and Apache 2.0 (code): (1) BankStress-330, the tiered benchmark, with per-query authoring notes and the full label-audit record (raw and audited labels); (2) the three native-reviewed transfer sets (Tamil $\times$2 registers, Telugu); and (3) the ablation/reranker harness and result-verification scripts, with all released per-query evaluation logs. The released bundle is a result-verification package that recomputes selected reported statistics from frozen per-query logs; it does not provide a fully self-contained end-to-end rerun of the complete pipeline, and the pipeline source, prebuilt indices, and fine-tuned reranker weights are not included (the released per-query rerank logs verify the reported reranker evaluation comparisons without the weights; the training script documents the training procedure but cannot regenerate the weights without the excluded pipeline configuration). The released result-verification package and its checksum manifest are available at `https://github.com/PiyushAgarwal9/nlu-wnut-artifacts`.

B Reproducibility and Compute Statement

All pipeline inference is CPU except the cross-encoder reranker and encoder fine-tuning, which used one NVIDIA RTX A4000 (16 GB), co-resident with an unrelated service throughout. Encoder adapter: LoRA rank 16, α=32, dropout 0.1 on query/value projections of multilingual-e5-base, triplet loss, batch 128, 6 epochs. Reranker fine-tune: 21,456 taxonomy-derived pairs, 1 epoch ($\sim$65 min). Arbiter: gpt-oss-20b served via a hosted API at temperature 0 with a fixed prompt and fixed decoding parameters; substitutes: Llama-3.1-8B-Instruct (same prompt and role) and bge-reranker-v2-m3 (568M). Encoder: intfloat/multilingual-e5-base. Full request parameters and per-call logs accompany the artifact where releasable. Wall-clock per 330-query evaluation run: $\sim$3 min with the arbiter, $<$1 min without; each 80-query transfer run $\sim$2 min. All nine selected headline statistics (flip rate, normalizer and arbiter discordances, holdout, shortlist recall, escalation rates, reranker pairwise, augmentation, and the Hinglish-TOP null) recompute from the released per-query records via `harness/verify_claims.py`; the bundle is a result-verification package; the pipeline source, prebuilt indices, and reranker weights are not part of the release (Appendix A).

Author Index

Agarwal, Piyush, 356
Agrawal, Aditya, 244
Ahmed, Khandaker Mamun, 210
Akca, Onat Dogan, 49
Anwar, Md Musfique, 26
Aßenmacher, Matthias, 72

Baloni, Raghvi, 65
Bernardin III, Frederick E., 182
Bhattacharjee, Ankit, 266
Bhaumik, Krityapriya, 266

Cardoza, Irvin, 332
Chambers, Nathanael, 167
Choi, Seongwoo, 313

Das, Souvik, 244
deJonge, Peter, 167
dela Cruz, Jane Arleth, 101
Dogan, Ilayda, 341

Eichstaedt, Johannes C, 286

Guo, Xin, 244

Haensch, Anna-Carolina, 72
Hanafusa, Kentaro, 217
Haralovic, Marko, 49
Hassan, Muhammad Mubashir, 1
Hendrickx, Iris, 101

Im, Sungkyun, 313
Islam, Nayeemul, 26

Kajiwara, Tomoyuki, 217
Kandlinger, Martin, 72
Kang, Hari, 313
Karakos, Damianos, 182
Klinger, Roman, 10
Kochedykov, Denis, 244

Larson, Martha, 101
Lee, Sangoh, 313
Li, Chak-Fai, 182
Li, Minsi, 49
Liu, Danni, 227
Liu, Timothy, 286
Lloret, Elena, 141

M, Sai Ganesh, 286
Ma, Bolei, 72
Macko, Dominik, 154
Maekawa, Daisuke, 217
Mendelsohn, Julia, 341
Moro, Robert, 154
Münker, Simon, 89

Narkedimilli, Sathwik, 286
Nguyen-Le, Phuong-Anh, 341
Ni, Kai, 244
Niehues, Jan, 227

Pappacoda, Gianmarco, 117
Park, Suhyeon, 313
Plum, Alistair, 89
Popp, Leonard, 227
Pozo, Paloma Moreda, 141
Pérez-Montero, Alba, 141

Rahbar, Saman, 325, 332
Rettinger, Achim, 65, 89
Rifat, S M Golam, 26
Rossouw, David, 332
Ruggeri, Federico, 117

Sarymsakova, Albina, 10
Sathvik, Msvpj, 286
Schwager, Nils, 89
See, Simon, 286
Shafin, Ashfaq Ali, 210
Sijan, Tanvir Ahmed, 26
Sinhamahapatra, Supriti, 227
Song, Sangmin, 313
Srba, Ivan, 154
Sung, Man-Ling, 182

Theune, Mariët, 49
Thomas, Philippe, 195
Torroni, Paolo, 117
Trautner, Denis, 195

Ulicny, Brian, 182

Vangapandu, Parmitha, 286

Yücetürk, Salih Eren, 49

Zhang, Ted, 244
Zhu, Xiliang, 332