GlobalNLP 2025

# Proceedings of the Workshop on Beyond English: Natural Language Processing for all Languages in an Era of Large Language Models

*associated with*

**The 15th International Conference on Recent Advances in Natural Language Processing RANLP'2025**

Edited by Sudhansu Bala Das, Pruthwik Mishra, Alok Singh,
Shamsuddeen Hassan Muhammad, Asif Ekbal and Uday Kumar Dasi

12 September, 2025
Varna, Bulgaria

The Workshop on Beyond English: Natural Language Processing
for all Languages in an Era of Large Language Models
Associated with the International Conference
Recent Advances in Natural Language Processing
RANLP'2025

**PROCEEDINGS**

Varna, Bulgaria
12 September 2025

Online ISBN 978-954-452-105-9

Designed by INCOMA Ltd.
Shoumen, BULGARIA

# Preface

The field of Natural Language Processing (NLP) has achieved remarkable progress in recent years, powered by the emergence of Large Language Models (LLMs) and generative AI. These advancements have significantly improved language technology for high-resource languages such as English, Mandarin, and German. However, the majority of the world's languages, including medium-resource, under-resourced, and low-resource ones—remain underserved due to limited datasets, resources, and linguistic tools.

The GlobalNLP 2025Workshop, titled "Beyond English: Natural Language Processing for All Languages in an Era of Large Language Models", was organized as part of RANLP 2025, held from 12 September 2025 in Varna, Bulgaria. This workshop provided an inclusive platform for researchers, linguists, developers, and practitioners worldwide to explore how cutting-edge NLP techniques can be extended to every language, regardless of its resource availability. We received a total of 28 paper submissions, of which a curated selection was accepted for inclusion in the proceedings following a rigorous peer-review process. The accepted papers span various domains, from cross-lingual modeling and corpus creation to NLP applications in healthcare, education, and cultural preservation. We were honored to have keynote talks by distinguished experts, including: The workshop featured two invited talks by distinguished researchers, offering complementary perspectives on multilingualism, machine translation, and the future of language technologies in medicine. The first invited speaker was Prof. Dipti Misra Sharma, Professor Emeritus at the International Institute of Information Technology (IIIT) Hyderabad, India. Her talk, titled "Multilingualism, LLMs, and Machine Translation", was delivered in the opening session from 09:00–09:45. Prof. Sharma is a pioneering figure in the field of Natural Language Processing, with a career spanning more than two decades of landmark contributions to machine translation, multilingual NLP, and linguistic resource development. At the Language Technologies Research Centre (LTRC) at IIIT Hyderabad, she has led large-scale government-funded initiatives to create corpora, treebanks, morphological analyzers, and evaluation frameworks that have become foundational resources for researchers worldwide. Her work bridges theoretical linguistics, computational modeling, and real-world deployment, with a strong focus on low-resource and morphologically rich languages. She has been instrumental in developing multilingual translation systems, interoperable linguistic tools, and pipelines for code-mixed language processing. Beyond her technical work, she has played a leading role in policy-level language technology planning in India and mentored a generation of NLP researchers. In her talk, Prof. Sharma traced the evolution of machine translation from its foundations to state-of-the-art approaches, with particular attention to the role of Large Language Models (LLMs). She emphasized both the opportunities and challenges of applying LLMs in multilingual contexts, highlighting their potential for linguistic inclusivity while underscoring the need for data-efficient, linguistically informed methods.

The second invited speaker was Prof. Michael G. Madden, Established Professor and leads the Machine Learning Research Group that he set up in 2001. His talk, "Advances in Natural Language Processing and Machine Learning for Medicine", was presented in the afternoon session from 13:25–14:10. Prof. Madden is an internationally recognized leader in machine learning, artificial intelligence, and data-driven modeling. Since founding the Machine Learning Research Group at Galway in 2001, he has produced influential work on deep learning, probabilistic reasoning, dynamic Bayesian networks, and reinforcement learning. His expertise is especially relevant to GlobalNLP 2025 through his focus on combining data-driven learning with structured background knowledge, a central challenge in adapting LLMs for specialised and multilingual applications. His career spans both academia and industry. He founded the AI spin-out company AnalyzeIQ Ltd, has served as a visiting scientist at leading institutions such as UC Berkeley, UC Irvine, and the University of Helsinki, and has fostered collaborations between academia, industry, and government. In his keynote, Prof. Madden showcased applications of NLP and machine learning in medicine, demonstrating how knowledge-aware and inclusive AI systems can be developed to support decision-making in sensitive, high-stakes domains. His talk highlighted the need for models that are both technically advanced and socially responsible, resonating strongly with the workshop's vision of inclusive global NLP.

In addition to paper sessions and keynote addresses, the workshop featured:

• A panel discussion on the challenges and opportunities in building multilingual LLMs.

• Interactive demo sessions showcasing NLP tools and technologies developed for diverse linguistic communities.

Core themes of the workshop included inclusivity, resource creation for under-represented languages, and the practical deployment of LLMs across domains such as education, healthcare, and cultural heritage.

• Data-efficient NLP: Transfer learning, few-shot and zero-shot methods for low-resource settings.

• Multilingual and cross-lingual modeling: Techniques adaptive to morphologically rich and typologically diverse languages.

• Semantic and ontology-driven approaches: Entity linking, semantic similarity, and knowledge graph integration.

• Resource creation and reuse: Development of sustainable corpora, tools, and evaluation benchmarks.

• Real-world impact: Applying NLP in domains such as education, healthcare, policy, and digital humanities.

• LLMs in practice: Deployments for code generation, document summarization, personalized conversational agents, and beyond.

We extend our deepest gratitude to all authors for submitting their research, and to the Program Committee members for their careful and insightful reviews. We especially thank our keynote speakers, panelists, and demonstrators for enriching the workshop with their expertise. Finally, we are thankful to the RANLP 2025 Organizing Committee for supporting and hosting this inclusive initiative. We hope that these proceedings will inspire ongoing research and collaboration toward more equitable and universal NLP.

GlobalNLP 2025 Organizing Committee

## Organizing Committee and Volume Editors

• Sudhansu Bala Das (Insight Research Ireland Centre for Data Analytics, University of Galway, Ireland)
• Pruthwik Mishra (SVNIT Surat, India)
• Alok Singh (University of Oxford, UK)
• Shamsuddeen Hassan Muhammad (AfricaNLP, Imperial College London)
• Asif Ekbal (IIT Jodhpur, India)
• Uday Kumar Das (Software Engineer, Dundalk)

## Program Committee

The program committee consists of distinguished researchers and practitioners from across the globe, actively working in the fields of Natural Language Processing (NLP) and Large Language Models (LLMs).
• Alexander Gelbukh (Instituto Politécnico Nacional, Mexico)
• Bidyut Kumar Patra (IIT BHU, India)
• Clarence Teo (Nanyang Technological University, Singapore)
• Gaurish Thakkar (University of Zagreb, Croatia)
• Helena Moniz (Universidade de Lisboa, Lisbon, Portugal)
• Idris Abdulmumin (DSFSI, University of Pretoria)
• Ibrahim Said Ahmad (Northeastern University)
• Juri Opitz (University of Zurich, Switzerland)
• Luan Thanh Nguyen (Vietnam National University Ho Chi Minh City, Vietnam)
• Marie-Aude Lefer (UCLouvain, Belgium)
• Mohammed Hasanuzzaman (Queen's University Belfast, UK)
• Moritz Schaeffer (Johannes Gutenberg University of Mainz, Germany)
• Muslim Jameel Sayed (Atlantic Technological University, Ireland)
• Pádraic Moran (University of Galway, Ireland)
• Paolo Rosso (Valencia Polytechnic University, Spain)
• Paul Buitelaar (University of Galway, Ireland)
• Soumik Mandal (NYU Tandon School of Engineering, USA)
• Surangika Ranathunga (Massey University, New Zealand)
• Uthayasanker Thayasivam (University of Moratuwa, Sri Lanka)

# Table of Contents

# Conference Program

# Towards the Creation of a Collao Quechua–Spanish Parallel Corpus Using Optical Character Recognition

**Gian Carlo Orcotoma** **Lida Leon** **Hugo Espetia**
Universidad Andina del Cusco
{016101111f, lleon, hespetia}@uandina.edu.pe

## Abstract

The Quechua language stands as a fundamental element of Peru's social and cultural identity, carries linguistic and cultural significance. However, it faces substantial challenges in terms of digital representation. One major limitation is the scarcity of resources such as a parallel corpus, which limits the development of technological resources for its analysis and practical application. This study addresses this gap through a methodology for building a parallel corpus using Optical Character Recognition (OCR). We digitized a collection of texts from a common origin to create a corpus that enables reliable access. The resulting corpus serves as a valuable asset for linguistic and Natural Language Processing (NLP) research, as well as for Quechua speakers. The source material derives from works produced by graduate students from the *Academia Mayor de la Lengua Quechua*, validated by academic staff, ensuring grammatical, syntactic and semantic integrity.

## 1 Introduction

Data from the 2017 Peruvian national census conducted by the National Institute of Statistics and Informatics (INEI) indicate that 13.6% of the Peruvian population identify themselves as Quechua speakers. The Cusco region has the fourth highest density of Quechua speakers, with 54.32% of its population who use Quechua as their first or secondary language in Peru.

Despite this, many government services are still not offered in Quechua, which makes it a crucial task to close this gap. In this study, we propose a technological approach using OCR to aid in the development of technology in this language.

Ortega et al. (2020) among other researchers in the field of natural language processing (NLP), categorize the Quechua language as Low Resource Language (LRL) due to the lack of available digital information. Technologies such as automatic translation, speech recognition, or natural language processing in general need a large amount of information and examples to be successfully trained.

The necessity for a parallel corpus made us think of the official website of Jehovah's Witnesses (https://www.jw.org/es/), who publish their magazines in almost 300 languages, including Quechua. This initially led to the idea of collecting this information using web scraping techniques. But most of these magazines, repositories, and websites had already been utilized in previous projects. In particular, the Jehovah's Witnesses website was comprehensively addressed in the JW300 project by Agić and Vulić (2019). Consequently we decided to get information from a different, yet equally reliable source.

We established contact with a Quechua language school based in Cusco city in Perú called Academia Mayor de la Lengua Quechua (AMLQ) which kindly granted us access to explore their library with an extensive collection of books. After reaching an agreement, we proposed a methodology for collecting the texts from this physical source. Our approach comprises several stages: i) Identification and collection of texts, ii) Photo environment setup, iii) Book digitizing, iv) Text labeling, v) Image preprocessing, vi) Text recognition and extraction, vii) Correction and evaluation of the OCR.

The importance of the result of this study lies in providing a methodology for the development of new technologies that require Collao Quechua - Spanish parallel corpora, such as automatic translators and sentence auto-completion systems. Furthermore, it contributes to the Quechua language preservation, by compiling and capturing examples of its usage in a digital medium that can be easily consulted.



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 1–6, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-001

## 2 Method

### 2.1 Identification and collection of texts

At the start of the project, the source of the texts was unclear. However, we had a clear starting point: Texts should be in a two-column format, aligned side by side in Quechua and Spanish, to facilitate the alignment and cleaning tasks in later stages.

At the end of their 8-month course, the students write a set of literary products, including stories, legends, poems, articles, academic essays, songs, etc. All of them are written in Quechua, with their respective side-by-side translation in Spanish and then printed in a book. After revision and correction by the school's own professors, the books are stored to be available in the school's library. The diversity in the book content arises from the many backgrounds of the students who learn this language. In a meeting for material identification, we counted approximately 30 pages per book and 10 books per year.

### 2.2 Photo environment setup

Following the identification of the texts, the procedure for digitizing the books was coordinated with the school. Ten books from the year 2021 were selected to be digitized; however, only five were finally processed with OCR.

The academy's library provided an ideal environment for capturing the photographs; The books are easily accessible and organized by year on shelves, and a central reading table is located in the available in the room. The photographs were taken on this table, using a Xiaomi Poco X3 Pro smartphone with a top-down angle for each page of the books.

To ensure optimal photo quality, and minimize the possibility of blurred or out of focus shots. We used a tripod, which stabilized the phone 20 cm above the book's pages, this distance was found to be the most suitable to capture the entire content of most pages. Some books were larger, so the tripod column needed to be raised a few centimeters to elevate the phone and increase the distance, in order to widen the field of view.

### 2.3 Book digitizing

Once the equipment was in place, the capture of the books began; maintaining consistent lighting and focus was essential to ensure image quality. To achieve this, the camera's "Pro" mode, which most phones have, was tested. While it is true that "Pro" mode has enough options to adjust camera exposure time, it does not have the option to numerically measure the focus distance. The Open Camera application was used as an alternative because it allowed these parameters to be adjusted and locked. The values used during capture were: exposure time: 1/50 s, aperture: f/1.79, ISO: 300 and focus distance: 20 cm (the same distance from the book to the phone's camera).

For the organization of the photographs, the prefix "AMLQ" was adopted. This prefix is configurable within the Open Camera application. The full file name of each photograph conforms to the format: AMLQ + date of the photograph + time of the photograph, for example: "AMLQ_20230103_123037.png". The photographs were stored in a folder created for each book, which was labeled with the year and the order in which the book was photographed. For example, "2021_L1", where "L" means "'Libro" (Book 1).

### 2.4 Photo labeling

A manual labeling approach was adopted, using the LabelImg tool, developed by Tzutalin (2024). Essentially, this involved enclosing the text areas on the pages within bounding boxes or rectangles. To distinguish between the two languages present, specific labels were applied: "que" label for areas containing Quechua text and "esp" label for those with Spanish text. We followed the YOLO standard described by Redmon et al. (2016) which utilizes four values (x, y, w, h) to define the center and size of each box in pixel units.

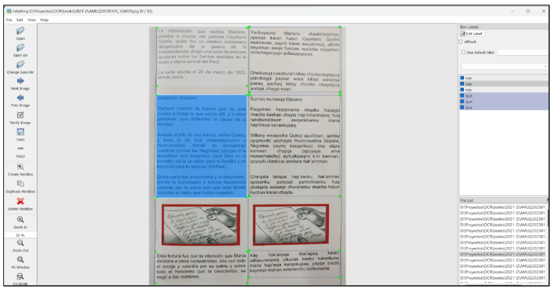


Figure 1: Labeling a page with Spanish and Collao Quecuha texts side by side with LabelImg software

### 2.5 Photo preprocessing

This step was developed using the Python programming language in conjunction with the OpenCV 2 library. The primary objective of this stage is twofold: first, to mitigate various errors that may arise from digitizing the physical books through photography, and second, to enhance the efficiency of the Tesseract OCR (Optical Character Recognition) system. To achieve these goals, two specific types of filters were applied to the captured images:

**Median Filter (Blur):** Its main purpose is to smooth the edges of the characters and reduce noise present in the images. It also helps to lessen the prominence of serifs often found in fonts like Times New Roman, which could potentially cause confusion during the OCR application process.

**Binarization Filter:** The aim of this filter is to transform the image into a black and white representation. This binary conversion is fundamental because it significantly facilitates text recognition and serves to eliminate variations in lighting that might exist within the image. For this specific case, the Otsu binarization algorithm, described by Otsu (1979), was chosen. Despite efforts to maintain consistent lighting during photography, slight variations did occur, But overall the Otsu algorithm is well-suited for handling images with such light variations.

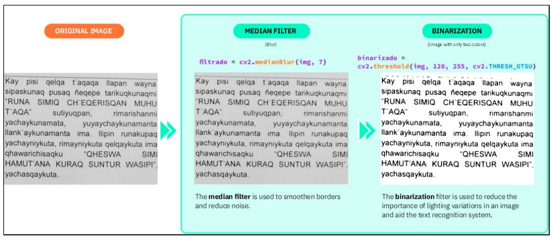



Figure 2: Photo preprocessing chain (median filter and binarization) software

## 2.6 Text recognition and extraction

Once the images are clean and ready, the text extraction process is carried out using Python and the Tesseract OCR engine. Tesseract was chosen due to its ease of use and its support for language recognition in both Quechua and Spanish which helped to minimize text recognition errors. Although the specific variety of Quechua supported by Tesseract OCR is unknown, it has proven to be better than the default spanish configuration in our photo database, accordingly, the corresponding configurations for both languages were used.

The output generated by the Tesseract OCR engine is in digital text format. For the purpose of this study, the Spanish and Quechua versions of the scanned text were saved in two separate .txt files. The naming convention for these files follows a defined pattern, it begins with the publication year of the book followed by the letter "L" to denote "libro" (book). Then, the order in which that book was published in that year is indicated. Finally, the language abbreviation of the content is specified ("esp" for Spanish or "que" for Quechua).

An example of this naming convention is "2021 l2_esp.txt" for the Spanish content of the second book from the 2021 set, and "2021 l2_que.txt" for its corresponding Quechua content. Each of these files contains the complete text of the book in the respective language. Additionally, within each file, the original pages are also separated and indicated using brackets to denote the page number, e.g.: [1] for page 1, [2] for page 2, etcetera.

## 2.7 Correction and evaluation of the OCR

Following the text recognition step, an analysis of the scanned texts revealed few scanning errors for Spanish. However, this was not the case with the Quechua language, where common character identification errors were detected, such as letter confusions (e.g., mistaking "q" for "g"), character duplications, and unrecognized characters.

To address these errors and effectively evaluate the proposed method, we took a sample from the corpus. Specifically, we chose the Quechua version scan of the second book from 2021 (2021_l2_que.txt) to generate a second corrected file which would serve as ground truth. This corrected version represents the most faithful transcription of the original text and was used as a reference to evaluate the accuracy of the OCR output.

To create this ground truth file, we collaborated with a Quechua professor from the Universidad Andina del Cusco. Utilizing the photos of the original book as a reference, the professor manually corrected the text extracted from the Quechua section of Book 2 from 2021 using Microsoft Word.

Once completed, the evaluation constituted the final step; we adopted the approach described by Rice (1996) who defines OCR quality evaluation as: "manipulation of character strings, which are transformed by an edit distance algorithm". Following a review of OCR evaluation methods and metrics presented in the work of Neudecker et al. (2021), we selected CER (Character Error Rate) and WER (Word Error Rate) metrics for evaluating the OCR accuracy. CER is calculated as follows:

$$\text{CER} = \frac{S + D + I}{N}$$

Where N is the total number of characters, I is the number of insertions, S is the number of substitutions, and D is the number of deletions needed in the OCR file in order to match the ground truth. WER is calculated similarly but at a word level.

For the evaluation, both CER and WER metrics were employed, as both represent the inverse precision of text recognition. After reviewing available tools, the open-source software ocrevaluation by Carrasco (2014), also described in the work of Neudecker et al. (2021) emerged as the most suitable option due to its comprehensive feature set, including the calculation of both CER and WER metrics, along with a comparative table of differences between the scanned text and the ground truth. ocrevaluation is available through a desktop Java application.

## 3 Results

Five graduation books from the Academia Mayor de la Lengua Quechua served as the data source for this study. These books contain texts in Spanish and Collao Quechua across diverse themes, including stories, poetry, history, science, lyrics, and personal narratives from the authors. To digitize the books, we set up an environment with uniform and constant lighting, a tripod, and a smartphone camera. Manual labeling and image preprocessing techniques were also employed to enhance the results of text recognition with the Tesseract OCR library. Subsequently, the text was stored in digital formats, which can be classified and located by year, book order, and language.

The corpus consists of a total of 44,263 words distributed across two languages. As shown in Table 1, the majority of the words are in Spanish, with 26,084 tokens (58.9%), while Quechua accounts for 18,179 tokens (41.1%). This distribution highlights the predominance of Spanish words in the dataset. However, it should be noted that the relatively lower word count in Quechua does not necessarily indicate less linguistic content, since Quechua is an agglutinative language in which a single word often carries the information that would require several words in Spanish.

| Language | Word count | Percentage |
|---|---|---|
| Spanish | 26,084 | 58.9% |
| Quechua | 18,179 | 41.1% |

Table 1: Word distribution by language

The scanned texts exhibit certain errors, such as the insertion of unwanted characters (e.g., punctuation marks, hyphens, and alphanumeric characters in incorrect positions throughout the corpus), character confusions (where one character is mistaken for a similar-looking one), and deletions or omissions of some characters. The recognized Quechua texts present these issues more frequently than the Spanish ones, preventing the corpus from being a 100% accurate reproduction of the original books.

For this reason, the Quechua text of book 2 from 2021 served as a sample to test the quality of the applied OCR. The evaluation, comparing the scanned text to the corrected text (or ground truth), revealed that 1.82% of characters were incorrectly detected according to the CER analysis, and 6.59% of words were incorrectly detected according to the WER analysis.

| | |
|---|---|
| **CER** | 1.82% |
| **WER** | 6.59% |
| **WER (order independent)** | 5.61% |

Table 2: CER and WER results

In addition to the Tesseract-based pipeline, we evaluated transformer-based OCR architectures, specifically TrOCR and DONUT, using the following pretrained models:

- microsoft/trocr-small-printed
- naver-clova-ix/donut-base-finetuned-cord-v2

TrOCR achieved satisfactory results for English text but consistently failed to recognize the Spanish and Quechua texts in the photos, producing incoherent outputs. This behavior is expected given that its base model lacks multilingual training for these languages. DONUT, on the other hand, recognized both Quechua and Spanish words, but failed to correctly identify the character “ñ” and produced substitutions, likely because this character was absent from its original vocabulary. However, it returned the output as a structured JSON object rather than plain text. This demonstrates its document understanding capability but also indicates the need for fine-tuning to align its output with the parallel corpus structure required in this work.

## 4 Discussion

The present study introduces a Collao Quechua - Spanish corpus along with the method employed for its construction. This corpus includes books from the Academia Mayor de la Lengua Quechua's library, featuring a broad spectrum of themes. This diversity contributes to the variability and richness of the corpus, making it suitable for future research.

To evaluate the quality of the method, the CER (Character Error Rate) and WER (Word Error Rate) metrics were calculated on a Quechua sample from the corpus, producing errors of 1.82% and 6.59%, respectively. These results, while revealing text recognition errors, are encouraging, especially considering that Tesseract's default configuration for Quechua was used. Such errors were anticipated, and many were mitigated thanks to the preprocessing step.

In the research made by Cordova and Nouvel (2021), the scope extended to digitizing and correcting a dictionary for the Ancash Quechua variant, in addition to training an OCR model adapted to the specificities of that material. Their work compared three OCR software programs, with Tesseract emerging as the most accurate; however, similar errors were observed. This suggests that default configuration precision is often insufficient for low-resource languages with numerous variants, such as the Quechua family.

The work of Agarwal and Anastasopoulos (2024) highlights that incorporating OCR adaptation stages for each particular case significantly improves text quality in languages with limited digital resources. For instance, Cordova and Nouvel (2021) trained their own OCR model, while the present work included a photo preprocessing phase. The quality and resolution of images, font type (handwritten or computerized), lighting, color, and other factors can drastically affect OCR results, therefore adapting each OCR method to the specific problem presented is important.

In this work, the labeling and post-OCR correction phases were performed manually. However, Agarwal and Anastasopoulos (2024) highlight the existence of automatic processes based on machine learning algorithms, which reduce manual labor and cost.

Transformer-based models like DONUT and TrOCR offer greater robustness and contextual understanding, compared to traditional OCR methods, yet they require adaptation and fine-tuning for Quechua. This represents a possible future development path for this project, given that the manual post-correction stage only covered the Quechua sample from Book 2, leaving the possibility of its application to the remaining books.

The corpus provides a comprehensive and thematically rich collection that will serve as a valuable resource for future research in NLP and linguistics for the Collao Quechua variant. It is worth reiterating that no post-OCR processing (cleaning) of the texts has been performed; addressing this problem surely presents an opportunity for future research.

## References


Milind Agarwal and Antonios Anastasopoulos. 2024. A concise survey of ocr for low-resource languages. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, page 88–102. Association for Computational Linguistics.

Željko Agić and Ivan Vulić. 2019. Jw300: A wide-coverage parallel corpus for low-resource languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Rafael C. Carrasco. 2014. An open-source ocr evaluation tool. In *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, DATeCH 2014, page 179–184. ACM.

Johanna Cordova and Damien Nouvel. 2021. Toward creation of ancash lexical resources from ocr. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, page 163–167. Association for Computational Linguistics.

Clemens Neudecker, Konstantin Baierer, Mike Gerber, Christian Clausner, Apostolos Antonacopoulos, and Stefan Pletschacher. 2021. A survey of ocr evaluation tools and metrics. In *The 6th International Workshop on Historical Document Imaging and Processing*, HIP '21, page 13–18. ACM.

John E. Ortega, Richard Castro Mamani, and Kyunghyun Cho. 2020. Neural machine translation with a polysynthetic low resource language. *Machine Translation*, 34(4):325–346.

Nobuyuki Otsu. 1979. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection.

Stephen Vincent Rice. 1996. *Measuring the accuracy of page-reading systems*. Ph.D. thesis.

Tzutalin. 2024. Labelimg. https://github.com/HumanSignal/labelImg. Accessed: 2025-06-03.

# Prompt Balance Matters: Understanding How Imbalanced Few-Shot Learning Affects Multilingual Sense Disambiguation in LLMs

**Deshan Sumanathilaka, Nicholas Micallef, Julian Hough**
Department of Computer Science, Swansea University, Wales, UK
{t.g.d.sumanathilaka, nicholas.micallef, julian.hough}@swansea.ac.uk

## Abstract

Recent advances in Large Language Models (LLMs) have significantly reshaped the landscape of Natural Language Processing (NLP). Among the various prompting techniques, few-shot prompting has gained considerable attention for its practicality and effectiveness. This study investigates how few-shot prompting strategies impact the Word Sense Disambiguation (WSD) task, particularly focusing on the biases introduced by imbalanced sample distributions. We use the GLOSSGPT prompting method, an advanced approach for English WSD, to test its effectiveness across five languages: English, German, Spanish, French, and Italian. Our results show that imbalanced few-shot examples can cause incorrect sense predictions in multilingual languages, but this issue does not appear in English. To assess model behavior, we evaluate both the GPT-4o and LLaMA-3.1-70B models and the results highlight the sensitivity of multilingual WSD to sample distribution in few-shot settings, emphasizing the need for balanced and representative prompting strategies.

## 1 Introduction

With the advent and rapid development of transformer architectures, Large Language Models (LLMs) have emerged as a game-changing technology for Natural Language Processing (NLP) tasks, particularly in text generation, question answering, and tasks requiring computational intelligence, reasoning, and language understanding (Minaee et al., 2024). Previous research has explored a variety of computational techniques in relation to LLMs, with a strong focus on prompt engineering, Retrieval Augmented Generation (RAG), knowledge base integration, and efficient fine-tuning strategies (Gu et al., 2024).

Among these areas, prompt engineering has received significant attention as a means of constructing accurate and efficient responses. Notably, few-shot prompting (Mann et al., 2020) has been extensively studied to enhance reasoning capabilities and in-context learning within prompting strategies.

Recent work, such as GLOSSGPT[1], has achieved state-of-the-art performance on the WSD task in English by leveraging few-shot prompting strategies. This approach demonstrates a strong ability to resolve lexical ambiguity (Sumanathilaka et al., 2025b). Other work has shown that zero-shot prompting alone cannot perform efficient WSD, but few-shot chain-of-thought (COT) can lead to higher-accuracy disambiguation (Sumanathilaka et al., 2024a). WSD remains a critical computational challenge for improving the understanding of word meanings when ambiguous terms appear in sentences or paragraphs. Effective WSD systems also contribute indirectly to advances in computational translation, transliteration, question answering and language understanding. While GLOSSGPT has demonstrated strong effectiveness for English, its generalizability to other languages remains unexplored. This research aims to address that gap by investigating whether the same approach can be effectively applied in a multilingual setup. In doing so, the study also examines how few-shot prompting may introduce bias into classification tasks such as WSD, specifically exploring whether models tend to favor high-frequency senses over low-frequency ones[2]. To analyze this behavior, we employ three sampling techniques namely Highest Frequency Sharing, Lowest Frequency Sharing, and Average Frequency Sharing as detailed in Section 3. Our findings and discussions are presented accordingly.

This study makes the following major contribu-

[1] https://github.com/Sumanathilaka/GlossGPT-GPT-4-WSD-with-COT

[2] Senses that are uncommon or rarely used



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 7–15, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-002

tions:

- We systematically investigate how different few-shot sampling strategies (Highest, Lowest, and Average Frequency Sharing) influence WSD performance across five languages. Our multilingual setup reveals that sense frequency imbalance introduces varying degrees of bias, with under-resourced languages being especially vulnerable.
- Our findings further highlight the importance of maintaining balanced few-shot examples as a critical factor for mitigating bias and improving disambiguation accuracy, especially in low-resource language contexts.
- We demonstrate that the optimal prompting strategy is language and model-specific, showing that a one-size-fits-all prompting approach fails to generalize effectively.

The remainder of the paper is organized as follows: Related Work, which discusses current research on multilingual WSD and few-shot bias studies; Methodology, which outlines the approach used to evaluate our proposed study; Results and Observations; and finally, Conclusions and Future Directions, which address potential strategies to mitigate bias in classification tasks across different language settings.

## 2 Related Work

We divide this section into two subsections. The first subsection describes WSD experiments in the context of Language Models (LMs) and LLMs, including recent advances. The second section describes experiments related to few-shot bias detection.

### 2.1 Advancements in Language models for WSD

Recent developments in language models have generated substantial interest in evaluating their performance across a range of NLP tasks. Sainz et al. (2023) demonstrated that LLMs possess an inherent ability to capture word senses, indicating their potential for WSD without explicit task-specific training. They framed WSD as a textual entailment task, prompting LLMs to assess the appropriateness of a domain label for a sentence containing an ambiguous word. Notably, this zero-shot approach surpassed random baselines and, in certain cases, matched or even outperformed supervised WSD systems (Ortega-Martín et al., 2023). Additionally, cross-lingual WSD has been explored through contextual word-level translation using pre-trained language models, with evaluations of zero-shot performance based on cross-lingual knowledge (Kang et al., 2023). A contrastive self-training framework, COSINE, was also proposed to fine-tune pre-trained LLMs using weak supervision without requiring labeled data (Yu et al., 2021). Manjavacas and Fonteyn (2022) investigated non-parametric learning approaches and fine-tuning strategies for LLMs applied to historical Dutch and English corpora. Qorib et al. (2024) highlighted the comparative effectiveness of encoder-only models over decoder-only architectures. Yae et al. (2024) examined the impact of LLM size on WSD performance, while Cahyawijaya et al. (2024) revealed limitations in cross-lingual WSD tasks, particularly involving false friends.[3]

Furthermore, Sumanathilaka et al. (2024a) demonstrated that prompt engineering techniques can significantly enhance WSD performance through in-context learning using GPT-3.5 Turbo and GPT-4-turbo. Their study explored various prompting strategies, including zero-shot, few-shot, and few-shot-CoT, highlighting the effectiveness of few-shot learning in improving sense prediction accuracy. It also showed that incorporating external knowledge further enhances the effectiveness of sense disambiguation. This work was further extended in subsequent studies, which showed that models such as Deepseek-R1 and o4-mini performed particularly well in WSD tasks compared to other flagship LLMs (Sumanathilaka et al., 2024b). These findings are also supported by Kibria et al. (2024). A key source of inspiration for this line of research is *GLOSSGPT* (Sumanathilaka et al., 2025b), which achieved state-of-the-art performance in English WSD by leveraging knowledge base-driven few-shot prompting. The model effectively incorporated lexical knowledge using WordNet glosses and synonyms. Although this approach outperforms several existing WSD systems, the direct impact of few-shot learning requires further investigation. To address this, we propose a sampling-based approach aimed at gaining deeper insights into how various few-shot configurations influence WSD performance across languages.

[3]Orthographically similar words that have entirely different meanings across languages

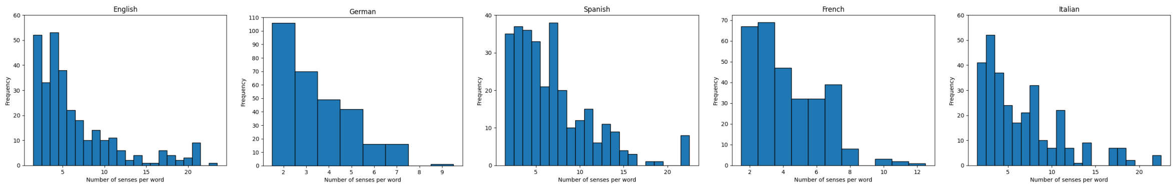



Figure 1: Sense distribution for selected samples on each language. The order is English, German, Spanish, French and Italian

### 2.2 Investigating Bias in Few-Shot Learning with LLMs

A very few recent studies have focused on the biases introduced by few-shot prompting in classification tasks, particularly in contexts involving LLMs. These biases often stem from prompt design, example selection, and label distribution, and can significantly affect model fairness and performance consistency.

The study by Lai et al. (2025) introduces a benchmark specifically for assessing short answer scoring with few-shot prompting. It highlights how LLMs amplify biases when prompted with limited, unbalanced examples and shows how model predictions become skewed toward overrepresented classes. Mallen and Belrose (2024) analyze the trade-off between label quantity and quality in few-shot prompts. These experiments reveal that weak labels often introduce substantial bias, especially in binary classification tasks. They also highlight that using a combination of low-quality and high-quality labels has a positive impact on the prediction process rather than either alone.

The study by Ma et al. (2023) revisits the problem of predictive bias, introducing a novel evaluation metric and proposing two algorithms namely T-fair Prompting and G-fair Prompting that aim to improve classification performance by selecting support examples that yield a more uniform distribution over output classes. More recently, Ahmadnia et al. (2025) emphasized that Few-Shot Learning performance degrades significantly when inappropriate support samples are selected. To address this, they introduced a new method that combines fine-tuning with Active Learning (AL) for support sample selection. Their approach leverages embedding techniques to extract salient features from unlabeled data and applies strategic sampling to select the most informative examples, thereby enhancing classification outcomes. Similarly, Pecher et al. (2024) highlighted the crucial role of sample quantity and quality in few-shot learning. Their work investigates how different sample selection strategies can be combined to mitigate the limitations posed by a restricted number of training examples and improve overall learning effectiveness.

These studies underscore the critical impact of few-shot prompting strategies on classification tasks, particularly emphasizing how imbalanced sample distributions can introduce predictive biases and affect both fairness and model reliability.

## 3 Methodology

This study has built upon a previously verified few-shot COT prompt provided by the GLOSSGPT. Prompts have been designed in English following a systematic chain of thought process, sequentially providing the lexical resources (gloss + synonyms) and a few possible few-shot instances extracted from pre-built KB, as illustrated in Figure 2. English prompts were used in all experiments, including multilingual setups, to minimize prompt ambiguity during inference. This ensures that the core evaluation remains focused on the WSD task itself, rather than being influenced by prompt design (Aina and Linzen, 2021). In this section, we present the dataset used, the techniques employed for knowledge base creation, and the sampling method applied for frequency sharing.

### 3.1 Datasets

#### 3.1.1 SemEval-2013 WSD dataset

For our evaluation, we use an updated version of SemEval-13[4], which contains four languages: Italian (IT), Spanish (ES), French (FR), and German (DE). English WSD was evaluated using the SemEval-13 English dataset (Jurgens and Klapaftis, 2013).

The sentence with multiple ambiguous words is split into different sentences, ensuring that each sentence contains only one ambiguous word, which

[4]https://github.com/SapienzaNLP/mwsd-datasets

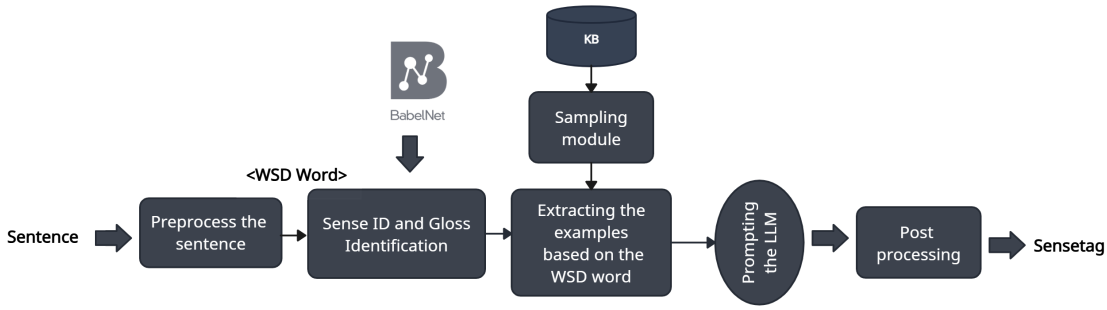



Figure 2: The data flow of the experiment process.

is enclosed between `<WSD>` tokens for the LLM inference task. A total of 300 random samples are utilized for the study for each language, with each sentence containing exactly one ambiguous word marked for disambiguation. A total of 300 samples were carefully selected to ensure that each ambiguous word had at least two distinct senses in BabelNet, a prerequisite for meaningful disambiguation. Priority was given to nouns, although samples also included other parts of speech (POS). The polysemy histogram for each language is shown in Figure 1. Furthermore, the study is constrained to a limited number of samples due to the practical limitations imposed by BabelNet's inference capabilities via API. To ensure consistency across evaluations, the same set of 300 random samples is used across all three sampling methods discussed in subsection 3.3. The micro F1 score is employed as the evaluation metric for assessing model performance.

#### 3.1.2 BabelNet

BabelNet (Navigli and Ponzetto, 2010) is the primary lexical knowledge base used for this study. It is a multilingual lexical and encyclopedic resource built by semi-automatically integrating various sources such as WordNet, multilingual WordNets, and Wikipedia. It contains multilingual synsets of synonymous terms across different languages, spanning 600 languages and includes over 23 million synsets. For this study, lexical knowledge resources are primarily obtained through the BabelNet API, which imposes a daily limit of 1,000 BabelCoins. This constraint necessitated limiting the study to a smaller set of samples. In addition, English lemmas and their corresponding synonyms are extracted from WordNet (Miller, 1995) to further enrich the lexical representation, especially in capturing and disambiguating ambiguous word meanings.

### 3.2 Knowledge-base creation for few-shot retrieval

The creation of the knowledge base (KB) was inspired by GLOSSGPT (Sumanathilaka et al., 2025b) and has been further enhanced to support a multilingual setup. The training data for all four languages is structured as a tree, with the language as the root node. The first-level parent nodes represent ambiguous words, the second-level nodes correspond to POS tags, and the child nodes contain example instances along with their respective BabelNet sense IDs. For efficient retrieval, the structure is stored in a JSON file. Based on the ambiguous word, the required information can be retrieved in constant time and shared with the model for few-shot prompting, following the sampling strategies described in subsection 3.3. A detailed structure is provided in Figure 3.

### 3.3 Sampling Strategies for Few-Shot Prompting in WSD

In this study, we apply few-shot prompting using the in-context learning paradigm to identify the correct sense of an ambiguous word in the WSD task. We explore how the frequency distribution of senses in the example pool affects the model's performance. Specifically, we define three sampling strategies based on the distribution of sense frequencies below: *Highest Frequency Sharing*, *Lowest Frequency Sharing*, and *Average Frequency Sharing*. We denote:

– $n$ as the total number of senses for a given ambiguous word,

| Word Sense | Meaning Description | Actual # | HF | LF | AF |
|---|---|---|---|---|---|
| `bank.n.14:00` | Financial institution | 7 | 7 | 1 | 4 |
| `bank.n.17:01` | Edge/slope of a river or body of water | 4 | 4 | 1 | 4 |
| `bank.n.17:00` | Raised embankment, like a ridge or mound | 1 | 1 | 1 | 1 |
| `bank.n.14:01` | Series/set (e.g., a bank of windows) | 1 | 1 | 1 | 1 |

Table 1: Sense Count and Example Report for the Word "Bank", according to the three frequency sharing techniques. H: High, L: Low, A: Average, F: Frequency

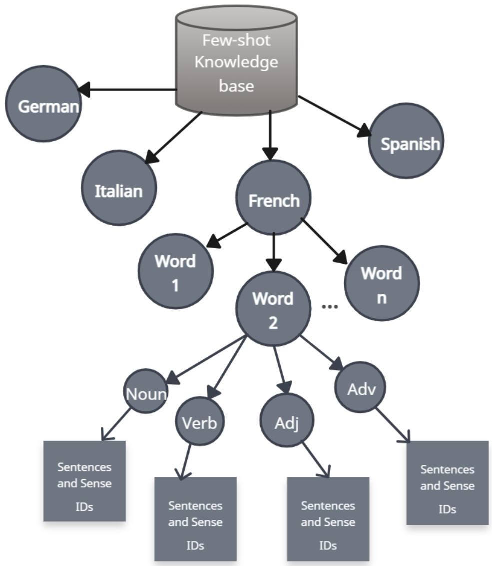



Figure 3: Few-shot knowledge base arrangement. For demonstration, the French branch is shown. A similar arrangement is followed for German, Italian and Spanish.

– freq($S_i$) as the number of available examples for sense $S_i$, where $i \in \{1, 2, ..., n\}$,

– $\mathcal{F} = \{\text{freq}(S_1), \text{freq}(S_2), \ldots, \text{freq}(S_n)\}$ as the set of all sense frequencies.

Each strategy defines how many few-shot examples $k_i$ are selected for each sense $S_i$, where $k_i \leq \text{freq}(S_i)$. Case selection is performed randomly when freq($S_i$) $> k_i$.

**1. Highest Frequency Sharing**

This strategy aims to balance the number of few-shot examples according to the most frequent sense. Each sense $S_i$ is assigned $k = \max(\mathcal{F})$ examples, if sufficient samples are available, else the number of samples available for that particular sense:

$$k_i = \min\left(\max(\mathcal{F}), \text{freq}(S_i)\right)$$

**2. Lowest Frequency Sharing**

This strategy equalizes the number of few-shot examples using the least frequent sense that has at least one sample. Each sense $S_i$ receives:

$$k_i = \min\left(\min(\mathcal{F} \setminus \{0\}), \text{freq}(S_i)\right)$$

This ensures that all senses are represented equally, without exceeding their available samples.

**3. Average Frequency Sharing**

This strategy computes an average of the minimum non-zero and maximum frequencies and uses it as a balanced number of examples per sense:

$$k = \left\lfloor \frac{\min(\mathcal{F} \setminus \{0\}) + \max(\mathcal{F})}{2} \right\rfloor$$
$$k_i = \min(k, \text{freq}(S_i))$$

This method serves as a compromise between the two extremes and reduces the effect of extreme imbalance.

Occasionally, it is observed that certain senses have no corresponding examples in the training data used to build the knowledge base for few-shot retrieval. In such cases, regardless of the sampling strategy applied, only the sense identifier (sense ID) is shared in the prompt, without any supporting examples. The frequencies and adjusted frequencies according to the three frequency sharing techniques is shown for *"Bank"* as a noun in Table 1.

### 3.4 Study Setup

For this study, we selected the GPT-4o and LLaMA 3.1-70B models due to their strong performance in multilingual settings (Vayani et al., 2025). General-purpose chat models were used in this study without any fine-tuning or prompt tuning. This was deliberately done to ensure the evaluation of the effectiveness of prompt engineering alone, using a few-shot example. Access to GPT-4o was obtained via the OpenAI API using a tier-one OpenAI account, while LLaMA-3.1-70B was accessed through the Together.ai API. Both models were configured with a temperature of 0 and a maximum output token limit of 500. The temperature selection for the study was inspired by previous work (Sumanathilaka et al., 2025a), and zero(0) was selected to ensure the deterministic responses of the study. The primary task assigned to both LLMs was word sense identification, with their role defined as a "*helpful assistant for identifying word senses*".

## 4 Results and Discussion

This section presents the results of our experiments, organized into three main areas of analysis. First, we evaluate the impact of different sampling strategies on performance. Next, we examine the suitability of various models for the task, highlighting their strengths and limitations. Finally, we explore the influence of contextual factors on the outcomes, providing insight into how context affects model behavior and overall system performance.

### 4.1 Effectiveness of Sampling Strategies

The experimental results in Table 2 show there is no universally optimal few-shot sampling strategy for WSD. The efficacy of any given strategy is highly context-dependent. *Average Frequency Sharing* often serves as a robust baseline, especially when paired with more capable models such as GPT-4o and applied to languages like English, German, and French in this study. Its balanced approach to sense representation generally proves beneficial, avoiding assigning too much weight to the high-frequency senses. *Highest Frequency Sharing* emerges as a specialized but highly effective strategy in certain linguistic contexts, specifically Spanish and Italian in this dataset, where it consistently outperforms other methods for both LLMs. This suggests that high-frequency examples can improve sense prediction in lower-resourced languages like Spanish and Italian.

In contrast, *Lowest Frequency Sharing* is generally a high-risk strategy, often resulting in sub-optimal or even the worst performance. Its occasional success appears to be tied to specific model-language combinations, for example, LLaMA 3.1 with English. However, the performance improvement is not significant compared to the average frequency sampling. The variability in optimal strategy across different conditions, such as model and language, highlights the importance of empirical evaluation when aiming for peak performance. In practice, WSD applications should test multiple strategies or base their choice on strong, evidence-backed reasoning that considers the characteristics of the LLM and target language. This also suggests the potential value of developing adaptive methods that can dynamically select or adjust sampling strategies based on context and sense distribution.

Overall, these results suggest that frequency-based sampling has limited influence on WSD performance in English, where even the lowest frequency sense for a word has sufficient samples for the model to attune to, but plays a more significant role in multilingual contexts, particularly for less resourced languages like Spanish and Italian.

### 4.2 Model-Specific Suitability for Few-Shot WSD

The choice of LLM significantly affects the effectiveness of few-shot sampling strategies. In this WSD study, GPT-4o generally achieved the highest performance overall. However, LLaMA 3.1 proved to be a strong competitor, even outperforming GPT-4o in certain cases, such as when using *Highest Frequency Sharing* in Spanish. Importantly, both models were sensitive to the sampling strategy used. Even a powerful model like GPT-4o did not perform best with a single strategy across all languages; for example, its performance in Spanish varied by 0.10 points depending on the sampling method.

This shows that model size or general capability alone cannot fully compensate for poor sampling choices. On the other hand, a model like LLaMA 3.1 can deliver excellent results when the sampling method is well-matched to its strengths and the task at hand. Conversely, thoughtful sampling design can improve results even for smaller or less advanced models. These results suggest that future work could benefit from developing model-aware sampling techniques.

### 4.3 Influence of Linguistic Context on Performance

Linguistic context has a significant impact on WSD performance. This study revealed a consistent performance hierarchy across the five languages examined: English and German >French >Spanish and Italian. This pattern held across both LLMs and most sampling strategies, indicating it

| Model | Method | English | German | Spanish | French | Italian |
|---|---|---|---|---|---|---|
| GPT-4o | Highest frequency | 0.81 | 0.76 | **0.70** | 0.75 | **0.74** |
| | Lowest frequency | 0.81 | 0.72 | 0.60 | 0.70 | 0.65 |
| | Average frequency | **0.83** | **0.78** | 0.64 | **0.76** | 0.70 |
| LLaMA-3.1 | Highest frequency | 0.75 | 0.76 | **0.73** | 0.72 | **0.68** |
| | Lowest frequency | **0.77** | 0.70 | 0.60 | 0.68 | 0.63 |
| | Average frequency | 0.76 | **0.77** | 0.65 | **0.74** | 0.66 |

Table 2: Performance comparison across languages for GPT 4o and LLaMA 3.1 under different frequency strategies. F1 scores are presented.

reflects deeper linguistic or resource-based differences rather than specific methodological choices.

Importantly, the optimal sampling strategy also varied by language. *Average Frequency Sharing* worked best for English, German, and French, while *Highest Frequency Sharing* was more effective for Spanish and Italian. This suggests that under-resourced languages like Spanish and Italian benefit from sampling strategies that emphasize more frequent and balanced sense representations to improve interpretation and disambiguation. The key takeaway is the need for language-aware WSD strategies. Achieving strong multilingual performance requires more than powerful models, which demands careful attention to each language's characteristics, including its sense distribution, resource availability, and representation in training data. This may involve tailored preprocessing, targeted resource development, or even fine-tuning models for specific languages or typological groups. A one-size-fits-all approach, typically optimized for English, is unlikely to perform well across the linguistic spectrum.

## 5 Conclusion and Future Directions

In conclusion, this research demonstrates that the selection of a few-shot examples in prompting LLMs can introduce significant performance variance in classification tasks in a multilingual setup, particularly when certain senses are overrepresented. However, in the case of English, such noticeable deviations are not identified. These results emphasize the importance of maintaining a balanced distribution of examples across all classes. The results also indicate that high-frequency sharing of sense examples can positively influence correct sense prediction, reinforcing the benefits of in-context learning during the inference process. Conversely, reducing few-shot examples to address class imbalance, especially for low-frequency senses, is not an effective strategy, as it can hinder the in-context learning process and degrade overall performance by limiting knowledge transfer. Imbalanced prompts tend to bias the model toward high-frequency senses, leading to reduced accuracy. While averaging techniques help mitigate such bias to some extent and contribute to more consistent performance, they are not a complete solution.

Overall, these findings underscore the need for balanced few-shot prompting with sufficiently rich examples to teach LLMs accurate sense disambiguation. These insights are particularly valuable when extending similar techniques to low-resource languages, where inherent limitations in language performance make balanced prompting even more critical. In such low-resourced multilingual adaptation setups, ensuring a well-balanced distribution of examples can significantly enhance both in-context learning and classification accuracy.

Future studies should focus on methodologies for balancing and improving few-shot learning, particularly in low-frequency and uncommon-sense scenarios. As suggested by Han et al. (2024); Li et al. (2024), multi-agent systems based on LLMs could be effectively utilized for context-aware few-shot generation, helping to create balanced examples necessary for the disambiguation process. These advancements can ensure that general-purpose LLMs are effectively leveraged for linguistic tasks such as WSD, rather than requiring fine-tuning for specific downstream applications.

The code and implementation are available at `https://github.com/Sumanathilaka/Prompt-Balance-Matters`.

## Limitations

One limitation of this study is that it considers only two flagship LLMs, which, while representative of current state-of-the-art performance, may not fully capture the diversity in model behavior. Although this does not compromise the strength of our findings, future evaluations with a broader range of models could provide further validation and insights. Additionally, the models used in this study are primarily chat-oriented; reasoning-focused models may exhibit different disambiguation capabilities, and evaluating such models would be a valuable extension. Another constraint is the limited sample size of 300 sentences per language. While this restricts the scale of the analysis, it ensures that each sampling technique operates on an identical and controlled dataset, thereby preserving consistency across evaluations without introducing bias from differing input distributions.


## Acknowledgments

This paper has been conducted in compliance with the ethical standards of Swansea University. Hough's work is supported by the EPSRC grant EP/X009343/1 'FLUIDITY'. Also, we would like to extend our sincere appreciation to the OpenAI Researcher Access Program for generously providing the credits that made the development of this project possible.


## References


Saeed Ahmadnia, Arash Yousefi Jordehi, Mahsa Hosseini Khasheh Heyran, Seyed Abolghasem Mirroshandel, Owen Rambow, and Cornelia Caragea. 2025. Active few-shot learning for text classification. *arXiv preprint arXiv:2502.18782*.

Laura Aina and Tal Linzen. 2021. The language model understood the prompt was ambiguous: Probing syntactic uncertainty through generation. *arXiv preprint arXiv:2109.07848*.

Samuel Cahyawijaya, Ruochen Zhang, Holy Lovenia, Jan Christian Blaise Cruz, Hiroki Nomoto, and Alham Fikri Aji. 2024. Thank you, stingray: Multilingual large language models can not (yet) disambiguate cross-lingual word sense. *arXiv preprint arXiv:2410.21573*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. 2024. Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*.

David Jurgens and Ioannis Klapaftis. 2013. Semeval-2013 task 13: Word sense induction for graded and non-graded senses. In *Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 290–299.

Haoqiang Kang, Terra Blevins, and Luke Zettlemoyer. 2023. Translate to Disambiguate: Zero-shot Multilingual Word Sense Disambiguation with Pretrained Language Models. ArXiv:2304.13803 [cs].

Raihan Kibria, Sheikh Dipta, and Muhammad Adnan. 2024. On functional competence of llms for linguistic disambiguation. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pages 143–160.

Peichao Lai, Kexuan Zhang, Yi Lin, Linyihan Zhang, Feiyang Ye, Jinhao Yan, Yanwei Xu, Conghui He, Yilei Wang, Wentao Zhang, et al. 2025. Sas-bench: A fine-grained benchmark for evaluating short answer scoring with large language models. *arXiv preprint arXiv:2505.07247*.

Xinyi Li, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. 2024. A survey on llm-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1):9.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. 2023. Fairness-guided few-shot prompting for large language models. *Advances in Neural Information Processing Systems*, 36:43136–43155.

Alex Mallen and Nora Belrose. 2024. Balancing label quantity and quality for scalable elicitation. *arXiv preprint arXiv:2410.13215*.

Enrique Manjavacas and Lauren Fonteyn. 2022. Non-parametric word sense disambiguation for historical languages. In *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, pages 123–134.

Ben Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, S Agarwal, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 1:3.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large Language Models: A Survey. ArXiv:2402.06196 [cs].

Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225.

Miguel Ortega-Martín, Óscar García-Sierra, Alfonso Ardoiz, Jorge Álvarez, Juan Carlos Armenteros, and Adrián Alonso. 2023. Linguistic ambiguity analysis in ChatGPT. ArXiv:2302.06426 [cs].

Branislav Pecher, Ivan Srba, Maria Bielikova, and Joaquin Vanschoren. 2024. Automatic combination of sample selection strategies for few-shot learning. *arXiv preprint arXiv:2402.03038*.

Muhammad Qorib, Geonsik Moon, and Hwee Tou Ng. 2024. Are decoder-only language models better than encoder-only language models in understanding word meaning? In *Findings of the Association for Computational Linguistics ACL 2024*, pages 16339–16347.

Oscar Sainz, Oier Lopez de Lacalle, Eneko Agirre, and German Rigau. 2023. What do Language Models know about word senses? Zero-Shot WSD with Language Models and Domain Inventories.

Deshan Sumanathilaka, Nicholas Micallef, and Julian Hough. 2024a. Assessing gpt's potential for word sense disambiguation: A quantitative evaluation on prompt engineering techniques. In *2024 IEEE 15th Control and System Graduate Research Colloquium (ICSGRC)*, pages 204–209. IEEE.

Deshan Sumanathilaka, Nicholas Micallef, and Julian Hough. 2025a. Exploring the impact of temperature on large language models: A case study for classification task based on word sense disambiguation.

Deshan Sumanathilaka, Nicholas Micallef, and Julian Hough. 2025b. GlossGPT: GPT for Word Sense Disambiguation using Few-shot Chain-of-Thought Prompting. *Procedia Computer Science*.

TGDK Sumanathilaka, Nicholas Micallef, and Julian Hough. 2024b. Can llms assist with ambiguity? a quantitative evaluation of various large language models on word sense disambiguation.

Ashmal Vayani, Dinura Dissanayake, Hasindri Watawana, Noor Ahsan, Nevasini Sasikumar, Omkar Thawakar, Henok Biadglign Ademtew, Yahya Hmaiti, Amandeep Kumar, Kartik Kukreja, et al. 2025. All languages matter: Evaluating lmms on culturally diverse 100 languages. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19565–19575.

Jung H Yae, Nolan C Skelly, Neil C Ranly, and Phillip M LaCasse. 2024. Leveraging large language models for word sense disambiguation. *Neural Computing and Applications*, pages 1–18.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2021. Fine-Tuning Pre-trained Language Model with Weak Supervision: A Contrastive-Regularized Self-Training Approach. ArXiv:2010.07835 [cs].

# Development of a Low-Cost Named Entity Recognition System for Odia Language using Deep Active Learning

**Tusarkanta Dalai**
NIT Rourklea
tusarkantadalai@gmail.com

**Tapas Kumar Mishra**
NIT Rourkela
mishrat@nitrkl.ac.in

**Pankaj K Sa**
NIT Rourkela
PankajKSa@nitrkl.ac.in

**Prithviraj Mohanty**
ITER, SOA University
prithvirajmohanty@soa.ac.in

**Chittaranjan Swain**
IIITM Gwalior
cswain@iiitm.ac.in

**Ajit Nayak**
ITER, SOA University
ajitnayak@soa.ac.in

## Abstract

Named Entity Recognition (NER) is a crucial component of Natural Language Processing (NLP) systems, which are utilized to extract significant information from massive quantities of unstructured textual data. The application of NER holds significant value in various NLP tasks, including but not limited to information retrieval, automatic question-answering systems, information extraction, and machine translation. Already, NER has accomplished fruitful achievements in English as well as in a number of other European languages. On the other hand, it is not yet well explored in Indian languages, primarily in the Odia language, remains insufficiently investigated owing to the absence of supporting tools and resources. In recent years, Machine Learning (ML) and deep learning (DL) based approaches have been able to achieve outstanding performance in constructing NLP tasks; nevertheless, these methods generally call for massive volumes of annotated corpus, which are costly to generate due to the need for domain-specific exports. Therefore, at present, researchers are utilizing the active learning approach, which involves the use of a sample selection technique in conjunction with supervised models. The aim of this approach is to minimize annotation expenses while optimizing the performance of ML and DL based models. The primary objective of this research is to develop a new active learning based NER system for Odia language.We applied a deep active learning (Deep-AL) strategy, and the deep active learning-based Odia NER system achieved nearly state-of-the-art performance. By utilizing only 38% of the original training data, we have achieved a maximum F1 score of 85.02%, which could save almost 62% of the cost for annotation.

## 1 Introduction

Identifying and classifying named entities (NE) into predefined classes such as person, location, organization, number, and time is a fundamental task in many NLP applications, commonly referred to as Named Entity Recognition (NER) (Nadeau and Sekine, 2007). Accurate identification of such entities is critical for extracting structured information from unstructured text. NER also plays a significant role in search engines for organizing, indexing, and linking named references consistently, thereby enhancing document searchability. For example, a NER system can help accurately determine individuals mentioned in news articles. Its utility has been well demonstrated in systems like Amazon Alexa and Apple Siri, particularly for Western languages. Moreover, NER is a key component in several downstream NLP tasks such as question answering, text summarization, machine translation (Bala Das et al., 2024, 2023; Das et al., 2025b,a), word-sense



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 16–26, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-003

disambiguation, coreference resolution, and semantic search.

Despite significant progress in European and many Asian languages, NER remains underexplored in low-resource languages, particularly Odia. Literature indicates that very limited attention has been paid to Odia in the context of NLP tasks, including NER. While Indian languages have seen increasing research in computational linguistics, Odia continues to lack the necessary tools and resources. Thus, it becomes essential to investigate NER from the lens of Odia language processing. Though a few studies on Odia NER exist, many of the resources used are either not publicly accessible or lack proper documentation. This work builds on earlier research in developing POS taggers for Odia (Dalai et al., 2023, 2024), and represents a step forward in advancing sequence labeling tasks for the language.

The development of a robust Odia NER system is crucial for enhancing Odia NLP applications. Several approaches such as probabilistic methods, rule-based systems, deep learning models, and hybrid strategies have been employed to address NER tasks. These systems aim to automatically tag entities in text. However, existing Odia NER systems mostly rely on conventional approaches like rule-based or machine learning techniques. The lack of Odia linguistic resources, including grammar knowledge and handcrafted features, poses a significant challenge. Additional linguistic complexities such as free word order, no capitalization, high ambiguity, and morphological richness further complicate NER development in Odia.

To address these challenges, deep learning (DL) methods are being increasingly adopted in NER system development. Popular models like CNNs, RNNs, LSTMs, and GRUs have shown success in other languages by leveraging multiple neural network layers to extract higher-level features. However, DL-based models have not yet been effectively applied to Odia NER. This work aims to bridge that gap by evaluating DL-based approaches for Odia. Since DL models require large volumes of annotated data, which are expensive and time-consuming to produce, we adopt an active learning (AL) approach. AL combines sample selection strategies with supervised learning to reduce annotation costs while maintaining model effectiveness.

AL is especially useful in scenarios where collecting large labeled datasets is not practical. As a semi-supervised technique, AL focuses on reducing manual labeling during training by iteratively selecting the most informative samples. In this study, we develop an active learning-based NER system for Odia using a relatively small annotated corpus. We employ a subset of the Odia NER dataset (Dalai et al., 2025), comprising 10,950 sentences annotated across twelve entity classes. The dataset and methodology are described in subsequent sections.

## 2 Related Work

This section presents a comprehensive overview of research and development in Named Entity Recognition (NER). The earliest significant effort in this area was introduced by (Grishman and Sundheim, 1996) at the Sixth Message Understanding Conference (MUC-6) in 1996, where the NER task focused on identifying entities such as persons, organizations, locations, percentages, and currency. Following this, numerous researchers contributed to the growth of the field (Sang and De Meulder, 2003; Demartini et al., 2009; Balog et al., 2010). Several advancements were later made in Indian language NER through rule-based approaches (Gupta and Lehal, 2011; Alfred et al., 2014; Riaz, 2010; Sasidhar et al., 2011). While such systems often yielded strong results, they had notable limitations, including high dependence on manual effort, slow learning capability, and substantial time requirements. Moreover, rule-based NER systems tend to be language-specific, making them difficult to adapt across different linguistic contexts. Due to these drawbacks, attention gradually shifted towards statistically-driven machine learning algorithms, which offered more flexibility and scalability for NER development.

The advancement of the NER system has encompassed the amalgamation of diverse statistical methodologies, such as the Support Vector Machine (SVM), Maximum Entropy (ME)(Saha et al., 2012), Hidden Markov Model (HMM)(Bikel et al., 1997; Morwal et al., 2012), Conditional Random Fields (CRF)(Mccallum, 2003), and other related techniques. These systems achieve this by integrating rule-based and ML-based approaches (Chopra et al., 2012; Biswas et al., 2010; Srivastava et al., 2011). Although machine learning-based NER systems exhibit remarkable performance, these systems nevertheless have a number of major disadvantages. These limitations include the requirement for extensive annotated datasets, the challenge of selecting an appropriate feature set, and the choice of an appropriate learning algorithm. Furthermore, researchers have initiated the development of DL models that avoid the need for traditional methodologies for the development of sequence labeling task.

Initially, (Collobert et al., 2011) devised an English NER model by utilizing characteristics acquired from word embeddings (WE) that were trained on an extensive collection of unlabeled data. (Chiu and Nichols, 2016) developed a NER system for the English language and this model incorporated both Bi-LSTM and CNN architectures to capture character-level details. In a similar manner, (Ma and Hovy, 2016) proposed a NER model for the English. The model, based on a combination of Bi-LSTM, CNN, and CRF, incorporates various deep learning techniques. In addition, (Athavale et al., 2016) developed a model for Hindi NER systems, which integrates pre-trained word embeddings with a Bi-LSTM architecture and a softmax layer. (Gupta et al., 2018) introduced an additional neural network model for NER that utilizes deep learning techniques. This model specifically focuses on code-mixed Indian social media content and employs a gated recurrent unit (GRU) along with character- and word-layer embeddings. However, the utilization of deep learning-based methods often necessitates a large quantity of annotated corpora. Nonetheless, the process of constructing such datasets demands a significant investment of time and extensive manual effort. Active learning has demonstrated promising results in situations where there is a limited corpus, thereby reducing the requirement for a large dataset. The system selects samples for labeling in an efficient manner. The active learning technique enables the algorithm to make informed decisions regarding the selection of instances for labeling, as opposed to the supervised learning mode, where a random subset of unlabeled instances is generated and labeled.

Many NLP applications, including information extraction (Settles and Craven, 2008), text classification (Tong and Koller, 2001), and word sense disambiguation (Zhu and Hovy, 2007), which need annotation from a huge pool of unannotated data to build a supervised ML model have benefited from Active Learning (AL) methodologies. However, the traditional AL algorithm fails to address high-dimensional data. Therefore, it is anticipated that the combination of active learning and deep learning will produce better results. Deep active learning has been employed extensively in numerous applications like text categorization (Schröder and Niekler, 2020; Zhang et al., 2017), image recognition (Gal et al., 2017; Gudovskiy et al., 2020), visual question answering (Lin and Parikh, 2017), and object detection (Aghdam et al., 2019; Feng et al., 2019).

A handful of NER model construction initiatives have been proposed for the Odia language. (Das and Patnaik, 2013) proposed the first Odia NER system; it made use of a support vector machine and attained an F1 score of 80% by feeding the feature set as language-specific rules, gazetteers, and context patterns. Following this, (Das et al., 2015) introduced an Odia NER system based on ML and trained on a manually annotated corpus of 1,000 sentences. For the purpose of data labeling, a set of ten tags was considered. This NER system achieved an F1 score of 81%. Subsequently, (Balabantaray et al., 2013) developed a NER system for the Odia language that was based on CRF, and they acquired an F1 score of 71%. In order to evaluate the effectiveness of the NER task, a variety of feature sets were generated using gazetteers and POS tags, respectively.

Based on our review of the relevant literature, we found that researchers have not paid much attention to Odia for NLP tasks such as NER, and only a small amount of study has been conducted on the language. Deep learning-based strategies were not utilized to their full potential when building the Odia Natural Language Engineering (NER) system.

## 3 System Model

### 3.1 Active Learning

This subsection describes the algorithmic procedure of Active Learning (AL), as depicted in Figure 1. The initial training samples for a machine learning or deep learning-based model are annotated by domain experts, and it picked according to a predefined strategy. After that, the annotated data is used to train the model; the unlabeled samples are ranked using a predetermined set of rules, and the best n samples are selected for annotation. Next, the annotated data are added to the training set, and the model is retrained using the updated training data. Iteration is performed on both the learning process and the selecting process up until the termination condition is met. It is very clear that the AL process ought to address the three significant concerns. The first step is the production of the initial training set, the second is the selection of an appropriate method for sample selection, and the third is the effective setup of the iterative process and the quit condition.

In this research, we modeled a real-world AL framework on pool-based resources. Even though we pre-annotated every sentence in our corpus, we did not make use their labels until the query algorithms picked them out.

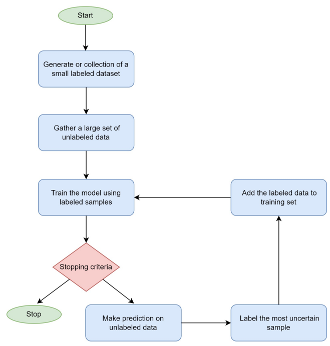



Figure 1: The process of active learning

Existing active learning methods have already demonstrated promising results in sequence labeling tasks. Three uncertainty-based strategies are implemented in our experiment: Least Confidence (LC), Bayesian Active Learning by Disagreements (BALD) and Maximum normalized log-probability (MNLP).

### 3.2 Deep Active Learning

This subsection presents a complete and methodical approach for the Deep active learning (Deep-AL)-based Odia NER system. DL has a high learning capacity when it comes to the processing of high-dimensional data and the automatic extraction of features, but AL has the ability to significantly minimize the costs associated with labeling the data. Consequently, it is clear to combine active learning and deep learning since this will considerably increase their applicability. Deep-AL was proposed by taking into account the combined benefits of the two methodologies. The framework of the Deep-AL model for the NER task of Odia language is illustrated in Figure 2. A deep learning model must first be initialized and pre-trained on labeled training data to extract features from unlabeled samples. After that, we chose samples by employing the corresponding query strategy, query the label by the manual annotator to construct a new training set, trained the deep learning model by making use of the updated training data, and then simultaneously updated the unlabeled pool. This method is repeated until the predetermined termination criteria are met. The Deep-AL architecture can, in its most basic form, be broken down into two parts: the AL query method applied to the unlabeled dataset and the DL model training procedure.

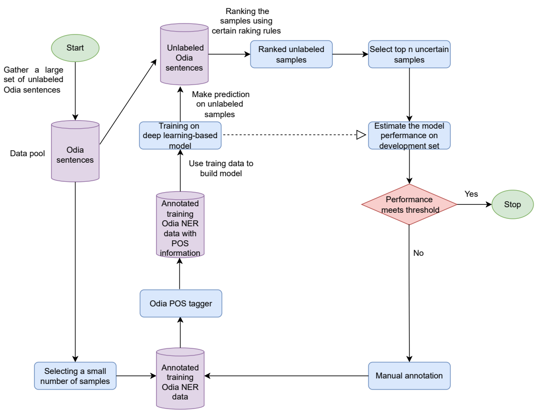



Figure 2: Framework of deep active learning model for Odia NER task

### 3.2.1 Deep Learning

The majority of AL approaches call for frequent retraining of the model when newly labeled instances are annotated. This is required in order to ensure optimal performance. As a consequence of this, it is essential that the model be capable of being retrained in a time-efficient manner. On the other hand, we would like to match with state-of-the-art deep learning-based models in terms of performance. In order to accomplish this, we must first determine the various deep learning architectures that comprise the Odia NER system. In this instance, a variety of DL-based models, including CNN, Bi-LSTM, models with CRF at inference layer are used to train the model. The architecture of the DL-based model for Odia NER system depicted in Figure 3. Figure 3 outlined the stages required in creating a DL model in order to make it simple.

1. The model takes an Odia sentence as input.
2. In order to incorporate information pertaining to the character sequences of Odia words, neural encoders such as CNN and Bi-LSTM models are employed as character-level embeddings.
3. Pre-trained FastText Odia word vector is used to initialize for word-level embeddings.
4. A fully connected NN is then fed the combined character and word embeddings.
5. The output of the previous layer gets inputted into the word sequence layer as input.
6. The output of the final hidden layer of the word sequence layer is utilized as input for the inference layer (CRF or softmax) in order to make predictions over possible tags associated with each input word.

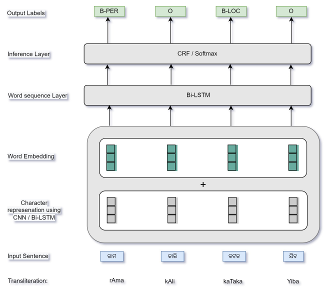



Figure 3: Architecture of Odia NER system using deep learning-based model

# 4 Experimental Results

## 4.1 Dataset Description

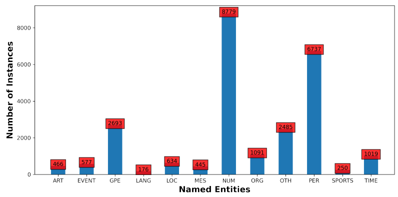
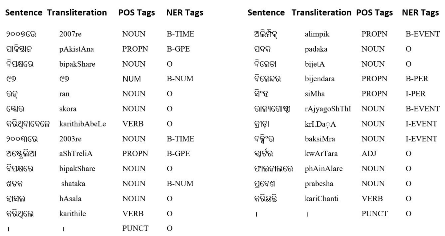



Figure 4: Class wise NEs in Odia NER dataset

For our experiment, we have used a subset of the public Odia NER corpus (Dalai et al., 2025). The Odia NER dataset includes the twelve NE types of PERSON, ORGANIZATION, GEOPOLITICAL ENTITY, LOCATION, EVENT, LANGUAGE, ART, SPORTS, NUMBER, TIME, MEASUREMENT, and OTHERS.



| Sentence | Transliteration | POS Tags | NER Tags |
|---|---|---|---|
| ୨୦୦୭ରେ | 2007re | NOUN | B-TIME |
| ପାକିସ୍ତାନ | pAkistAna | PROPN | B-GPE |
| ବିପକ୍ଷରେ | bipakShare | NOUN | O |
| ୯୭ | ୯୭ | NUM | B-NUM |
| ରନ୍ | ran | NOUN | O |
| ସ୍କୋର | skora | NOUN | O |
| କରିଥିବାବେଳେ | karithibAbeLe | VERB | O |
| ୨୦୦୩ରେ | 2003re | NOUN | B-TIME |
| ଅଷ୍ଟ୍ରେଲିଆ | aShTreliA | PROPN | B-GPE |
| ବିପକ୍ଷରେ | bipakShare | NOUN | O |
| ଶତକ | shataka | NOUN | B-NUM |
| ହାସଲ | hAsala | NOUN | O |
| କରିଥିଲେ | karithile | VERB | O |
| । | । | PUNCT | O |

| Sentence | Transliteration | POS Tags | NER Tags |
|---|---|---|---|
| ଅଲିମ୍ପିକ୍ | alimpik | PROPN | B-EVENT |
| ପଦକ | padaka | NOUN | O |
| ବିଜେତା | bijetA | NOUN | O |
| ବିଜେନ୍ଦର | bijendara | PROPN | B-PER |
| ସିଂହ | siMha | PROPN | I-PER |
| ରାଜ୍ୟଗୋଷ୍ଠୀ | rAjyagoShThI | NOUN | B-EVENT |
| କ୍ରୀଡ଼ା | krI.Da଼A | NOUN | I-EVENT |
| ବକ୍ସିଂର | baksiMra | NOUN | I-EVENT |
| କ୍ୱାର୍ଟର | kwArTara | ADJ | O |
| ଫାଇନାଲରେ | phAinAlare | NOUN | O |
| ପ୍ରବେଶ | prabesha | NOUN | O |
| କରିଛନ୍ତି | kariChanti | VERB | O |
| । | । | PUNCT | O |

Figure 5: A sample of Odia NER dataset

The dataset included 10,950 annotated sentences with a total of 158,947 tokens and 25,352 named entities. Figure 4 displays the statistics of the named entities in the Odia NER dataset. The dataset is split into three distinct parts: (1) the development set; (2) the training set that will be queried; and (3) the test set that will be evaluated. The distribution of the Odia NER corpus is shown in Table 1.

Table 1: **Odia NER corpus details**

| Data | Number of Sentences | Number of tokens |
|---|---|---|
| Training | 7660 | 1,11,250 |
| Testing | 1650 | 23,830 |
| Development | 1640 | 23,867 |
| Total | 10,950 | 1,58,947 |

## 4.2 Results

In order to verify the effectiveness and performance of our Deep-AL model, we have implemented different deep-learning techniques. Training the Odia NER models involved the usage of the Bi-LSTM classifier, which encoded words using a Bi-LSTM model and character level encoding using either of CNN or Bi-LSTM, and finally, the inference layer was handled by softmax or CRF tag decoder. We employed the conventional separation of the datasets, which included training, validation, and test data. Our Odia NER dataset was divided according to the usual 70% / 15% / 15% split, with 70% going to training and 15% each to validation and testing. The performance of the test dataset is used to determine parameters such as the number of iterations, learning rate, etc. To initialize the token, we employed a character embedding size of 30 and word embedding size of 300. Our complete deep learning system was trained with a stochastic gradient descent optimizer with a learning rate of 0.001, batch size of 128, and dropout rate of 30%. Our model comprised 300-dimensional word embeddings (WE) and utilized the pretrained FastText model. We trained the models for 30 epochs. The number of active learning iterations was set at 25 due to the observation that each algorithm does not exhibit significant improvement after 20 iterations.

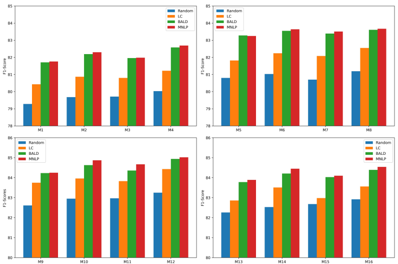


Figure 6: F1-Score of different models with active learning strategies, M1: Bi-LSTM + Softmax, M2: Bi-LSTM + Softmax + POS , M3: Bi-LSTM + CRF , M4: Bi-LSTM + CRF + POS , M5: WE + Bi-LSTM + Softmax , M6: WE + Bi-LSTM + Softmax +POS , M7: WE + Bi-LSTM + CRF , M8: WE + Bi-LSTM + CRF + POS , M9: WE + CharCNN + Bi-LSTM + Softmax , M10: WE + CharCNN + Bi-LSTM + Softmax + POS, M11: WE + CharCNN + Bi-LSTM + CRF, M12: WE + CharCNN + Bi-LSTM + CRF + POS , M13: WE + CharBi-LSTM + Bi-LSTM + Softmax , M14: WE + CharBi-LSTM + Bi-LSTM + Softmax + POS , M15: WE + CharBi-LSTM + Bi-LSTM + CRF , M16: WE + CharBi-LSTM + Bi-LSTM + CRF + POS.

The Deep-AL process begins with a random selection of samples from the training dataset, on which the model was trained. Following this, the learning process consists of numerous iterations. At the beginning of each round, the Deep-AL algorithm selects the unannotated sentences from the data pool to be annotated based on a specified budget. After the samples are labeled, they are incorporated into the training data, and the data pool and training set is then updated. Therefore, the model parameters are modified through training on the current training dataset before proceeding to the next iteration. We initiated our experiments with 2% of the training data from the Odia NER corpus that was labeled. In addition, the same number of data was added at each learning iteration, and the precision, recall, and F1 score are used to evaluate the model performance on the testing set. Furthermore, we detailed the performance of our model following its completion of training. Each experiment was repeated five times, and the average F-scores are recorded. The results depicted in Figure 7 demonstrate that all active learning algorithms outperform the random baseline in the Odia NER corpus. Additionally, the results indicate that the MNLP approach displays superior performance when compared to other active learning strategies on Odia NER dataset.

Table 2 depicted the results of our comparative analysis of the Odia NER performance of several models with MNLP active learning strategy. The graph depicted in Figure 7 displays F1 scores on the y-axis and the proportion of tagged words used for training on the x-axis. The results indicate that active learning methods utilizing only 38% of the training data on the Odia NER dataset were able to achieve 99% of the performance of the deep learning model that was trained with complete data. Table 3 presents the precision, recall, and F1 score for each distinct named entity class in our optimal Odia named entity recognition system.

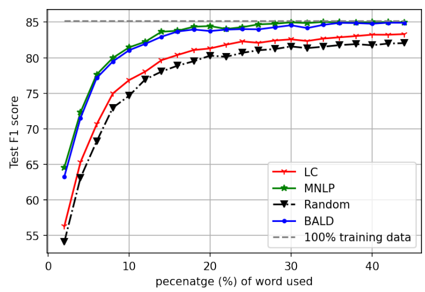



Figure 7: F1 score on the test dataset, in terms of the number of words labeled

Table 2: **Precision, Recall and F1 score of different models on Test data**

| Model | Usage of POS information | Precision | recall | F1-Score |
|---|---|---|---|---|
| Bi-LSTM + Softmax | NO | 82.67 | 80.87 | 81.76 |
| Bi-LSTM + CRF | | 82.85 | 81.07 | 81.98 |
| WE + Bi-LSTM + Softmax | | 84.23 | 82.29 | 83.25 |
| WE + Bi-LSTM + CRF | | 84.56 | 82.48 | 83.51 |
| WE + CharCNN + Bi-LSTM + Softmax | | 85.11 | 83.40 | 84.25 |
| WE + CharCNN + Bi-LSTM + CRF | | 85.36 | 84.00 | 84.67 |
| WE + CharBi-LSTM + Bi-LSTM + Softmax | | 85.03 | 82.78 | 83.89 |
| WE + CharBi-LSTM + Bi-LSTM + CRF | | 85.18 | 83.05 | 84.10 |
| Bi-LSTM + Softmax | YES | 83.28 | 81.34 | 82.30 |
| Bi-LSTM + CRF | | 83.92 | 81.49 | 82.69 |
| WE + Bi-LSTM + Softmax | | 84.63 | 82.68 | 83.64 |
| WE + Bi-LSTM + CRF | | 84.55 | 82.80 | 83.67 |
| WE + CharCNN + Bi-LSTM + Softmax | | 85.29 | 84.45 | 84.87 |
| WE + CharCNN + Bi-LSTM + CRF | | 85.76 | 84.29 | **85.02** |
| WE + CharBi-LSTM + Bi-LSTM + Softmax | | 85.11 | 83.80 | 84.45 |
| WE + CharBi-LSTM + Bi-LSTM + CRF | | 85.23 | 83.86 | 84.54 |

Table 3: **Label wise score of WE+CharCNN+Bi-LSTM+CRF model on Test data**

| Name entity | Precision | Recall | F1-score |
|---|---|---|---|
| ART | 87.16 | 82.92 | 84.99 |
| EVENT | 81.29 | 78.36 | 79.80 |
| GPE | 91.90 | 89.63 | 90.75 |
| LANG | 90.32 | 90.11 | 90.21 |
| LOC | 79.51 | 75.79 | 77.61 |
| MES | 92.32 | 94.56 | 93.43 |
| NUM | 91.78 | 93.97 | 92.47 |
| ORG | 82.18 | 76.39 | 79.18 |
| OTH | 82.37 | 77.46 | 79.84 |
| PER | 89.40 | 91.62 | 90.50 |
| SPORTS | 66.67 | 71.36 | 68.94 |
| TIME | 94.29 | 89.43 | 91.80 |
| Macro average | 85.76 | 84.29 | 85.02 |

## 5 Conclusion and Future Work

In this work, we presented a cost-effective and resource-efficient NER system for the low-resource Odia language using a Deep-AL framework. By integrating deep learning architectures with active sample selection strategies, we addressed the challenges posed by limited annotated data, high labeling costs, and the linguistic complexity of Odia. Our proposed approach demonstrated that high performance can be achieved up to an F1 score of 85.02% using only 38% of the annotated data required by traditional deep learning models, thereby reducing annotation costs by approximately 62%. Through extensive experimentation, we showed that incorporating character-level features, pretrained FastText embeddings, POS information, and a CRF-based inference layer led to improved model performance. The results indicate that our approach not only outperforms standard supervised methods but also demonstrates scalability and efficiency, making it suitable for similar low-resource language settings. Despite promising outcomes, there are several directions for future enhancement of this work: The framework can be adapted for other low-resource Indian languages, facilitating cross-lingual and multilingual NER systems with shared architectures and embeddings. Future experiments may involve incorporating transformer-based architectures such as BERT, XLM-R, or IndicBERT for richer contextual representation and better generalization. Although we evaluated common strategies like LC, BALD, and MNLP, future work could explore more advanced, or hybrid query strategies tailored specifically for NER in agglutinative and morphologically rich languages like Odia. This research represents a significant step toward democratizing NLP technology for low-resource languages and highlights the practical feasibility of deploying scalable, accurate NER systems under constrained resources.

## References


Hamed H Aghdam, Abel Gonzalez-Garcia, Joost van de Weijer, and Antonio M López. 2019. Active learning for deep detection neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3672–3680.

Rayner Alfred, Leow Chin Leong, Chin Kim On, and Patricia Anthony. 2014. Malay named entity recognition based on rule-based approach. *International Journal of Machine Learning and Computing*, 4(3).

Vinayak Athavale, Shreenivas Bharadwaj, Monik Pamecha, Ameya Prabhu, and Manish Shrivastava. 2016. Towards deep learning in hindi ner: An approach to tackle the labelled data sparsity. In *Proceedings of the 13th International Conference on Natural Language Processing*, pages 154–160.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2023. Improving multilingual neural machine translation system for indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–24.

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, and Asif Ekbal. 2024. Multilingual neural machine translation for indic to indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(5):1–32.

Rakesh Balabantaray, Suprava Das, and Kshirabdhi Tanaya Mishra. 2013. Case study of named entity recognition in odia using crf++ tool. *International Journal of Advanced Computer Science and Applications*, 4(6).

Krisztian Balog, Pavel Serdyukov, and Arjen P de Vries. 2010. Overview of the trec 2010 entity track. Technical report, NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM.

Daniel M Bikel, Scott Miller, Richard Schwartz, and Ralph Weischedel. 1997. Nymble: a high-performance learning name-finder. In *Fifth Conference on Applied Natural Language Processing*, pages 194–201.

S Biswas, MK Mishra, S Acharya Sitanath_biswas, and S Mohanty. 2010. A two stage language independent named entity recognition for indian languages. *IJCSIT) International Journal of Computer Science and Information Technologies*, 1(4):285–289.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the association for computational linguistics*, 4:357–370.

Deepti Chopra, Nusrat Jahan, and Sudha Morwal. 2012. Hindi named entity recognition by aggregating rule based heuristics and hidden markov model. *International Journal of Information*, 2(6):43–52.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

Tusarkanta Dalai, Anupam Das, Tapas Kumar Mishra, and Pankaj Kumar Sa. 2025. Odner: Ner resource creation and system development for low-resource odia language. *Natural Language Processing Journal*, 11:100139.

Tusarkanta Dalai, Tapas Kumar Mishra, and Pankaj K Sa. 2023. Part-of-speech tagging of odia language using statistical and deep learning-based approaches. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*

Tusarkanta Dalai, Tapas Kumar Mishra, and Pankaj K Sa. 2024. Deep learning-based pos tagger and chunker for odia language using pretrained transformers. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(2):1–23.

Bishwa Das and Srikanta Patnaik. 2013. Name entity recognition for odia language using support vector machine.

Bishwa Ranjan Das, Srikanta Patnaik, Sarada Baboo, and Niladri Sekhar Dash. 2015. A system for recognition of named entities in odia text corpus using machine learning algorithm. In *Computational Intelligence in Data Mining-Volume 1*, pages 315–324. Springer.

Sudhansu Bala Das, Samujjal Choudhury, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025a. Investigating the effect of backtranslation for indic languages. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 152–165.

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025b. Statistical machine translation for indic languages. *Natural Language Processing*, 31(2):328–345.

Gianluca Demartini, Tereza Iofciu, and Arjen P de Vries. 2009. Overview of the inex 2009 entity ranking track. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 254–264. Springer.

Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. 2019. Deep active learning for efficient training of a lidar 3d object detector. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 667–674. IEEE.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.

Ralph Grishman and Beth M Sundheim. 1996. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. 2020. Deep active learning for biased datasets via fisher kernel self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9041–9049.

Deepak Gupta, Asif Ekbal, and Pushpak Bhattacharyya. 2018. A deep neural network based approach for entity extraction in code-mixed indian social media text. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Vishal Gupta and Gurpreet Singh Lehal. 2011. Named entity recognition for punjabi language text summarization. *International journal of computer applications*, 33(3):28–32.

Xiao Lin and Devi Parikh. 2017. Active learning for visual question answering: An empirical study. *arXiv preprint arXiv:1711.01732*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Andrew Mccallum. 2003. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proc. of CoNLL, 2003*.

Sudha Morwal, Nusrat Jahan, and Deepti Chopra. 2012. Named entity recognition using hidden markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol*, 1.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Kashif Riaz. 2010. Rule-based named entity recognition in urdu. In *Proceedings of the 2010 named entities workshop*, pages 126–135.

Sujan Kumar Saha, Pabitra Mitra, and Sudeshna Sarkar. 2012. A comparative study on feature reduction approaches in hindi and bengali named entity recognition. *Knowledge-Based Systems*, 27:322–332.

Erik 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.

B Sasidhar, PM Yohan, A Vinaya Babu, and A Govardhan. 2011. Named entity recognition in telugu language using language dependent features and rule based approach. *International Journal of Computer Applications*, 22(8):30–34.

Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks. *arXiv preprint arXiv:2008.07267*.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079.

Shilpi Srivastava, Mukund Sanglikar, and DC Kothari. 2011. Named entity recognition system for hindi language: a hybrid approach. *International Journal of Computational Linguistics (IJCL)*, 2(1):10–23.

Simon Tong and Daphne Koller. 2001. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66.

Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Jingbo Zhu and Eduard Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 783–790.

# Non-Contextual BERT or FastText? A Comparative Analysis

**Abhay Shanbhag[1,3], Suramya Jadhav[1,3], Amogh Thakurdesai[1,3], Ridhima Sinare[1,3] and Raviraj Joshi[2,3]**
[1]Pune Institute of Computer Technology, Pune
[2]Indian Institute of Technology Madras, Chennai
[3]L3Cube Labs, Pune

## Abstract

Natural Language Processing (NLP) for low-resource languages, which lack large annotated datasets, faces significant challenges due to limited high-quality data and linguistic resources. The selection of embeddings plays a critical role in achieving strong performance in NLP tasks. While contextual BERT embeddings require a full forward pass, non-contextual BERT embeddings rely only on table lookup. Existing research has primarily focused on contextual BERT embeddings, leaving non-contextual embeddings largely unexplored. In this study, we analyze the effectiveness of non-contextual embeddings from BERT models (MuRIL and MahaBERT) and FastText models (IndicFT and MahaFT) for tasks such as news classification, sentiment analysis, and hate speech detection in one such low-resource language—Marathi. We compare these embeddings with their contextual and compressed variants. Our findings indicate that non-contextual BERT embeddings extracted from the model's first embedding layer outperform FastText embeddings, presenting a promising alternative for low-resource NLP.

## 1 Introduction

Word embedding is a way of representing words into dense vectors in a continuous space such that the vectors capture the semantic relationship between the words for the models to understand the context and meaning of the text. FastText, a context-independent method, basically captures the subword information, enabling it to learn rare words, misspelled words, and out-of-vocabulary words. It is recognized in the NLP community for its efficient performance in tasks like text classification and sentiment analysis. Despite being relatively old, it still remains one of the most effective alternatives when performing tasks on large datasets across various languages due to its subword-based approach.

BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2018) word embeddings understand the meaning of a word based on its context in a sentence. The embeddings extracted just before the first embedding layer of the BERT architecture are referred to as non-contextual embeddings, while those obtained from the last hidden layer of BERT are known as contextual embeddings (Refer Figure 1). Numerous variations of BERT like IndicBERT (Kakwani et al., 2020), MuRIL (Khanuja et al., 2021), AfriBERT (Ralethe, 2020), and mBERT (Devlin et al., 2018) to name a few, are available for experiments.

Recent studies have experimented with both FastText and BERT for various tasks; however, most of them focus on exploring contextual BERT embeddings. Experiments of D'Sa et al. (2020) demonstrated that BERT embeddings outperformed FastText for classifying English text into toxic and non-toxic. Findings of Ahmed et al. (2024) suggested that BERT embeddings outperformed those of FastText with an F1 score of 84% when evaluated for depressive post-detection in Bangla.

While BERT consistently outperforms other word embeddings in various tasks for high-resource languages (HRLs) like English (Malik et al. (2021)), its effectiveness in low-resource languages (LRLs) remains relatively underexplored. This gap is particularly pronounced when balancing model performance with computational efficiency, which becomes a critical factor in low-resource settings.

Previous studies (D'Sa et al. (2020)) have focused on contextual BERT embeddings, which outperform FastText due to their ability to capture contextual information. However, the use of non-contextual BERT embeddings for classification tasks in low-resource languages like Marathi remains unexplored. Unlike contextual embeddings, which require a full forward pass through the model, non-contextual embeddings can be obtained through a simple table lookup. To our knowledge,



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 27–33, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-004

no prior work has examined the effectiveness of non-contextual BERT embeddings. We investigate how these embeddings, extracted from the model's first layer, compare to FastText embeddings for tasks such as news classification, sentiment analysis, and hate speech detection in Marathi.

Additionally, past comparisons often used BERT's 768-dimensional embeddings against FastText's 300-dimensional ones, which is unfair since higher dimensions naturally provide better feature extraction. To address this, we ensure a fair comparison by reducing the BERT embeddings to 300 dimensions.

This paper focuses on utilizing FastText and non-contextual BERT for the Marathi language for the following tasks: Sentiment Classification, 2-Class and 4-Class Hate Speech Detection, and News Article Classification for headlines, long paragraphs, and long documents. We construct a comprehensive analysis of FastText embeddings, IndicFT (Kakwani et al., 2020) and MahaFT (Joshi, 2022) embeddings, and BERT embeddings, including muril-base-cased (Khanuja et al., 2021) and marathi-bert-v2 (Joshi, 2022). To enhance the comparison, we replicate the experiments using widely utilized contextual BERT embeddings. We also evaluate the impact of compression on both contextual and non-contextual BERT-based embeddings. Our analysis shows that non-contextual BERT embeddings generally perform better than FastText in most tasks. Furthermore, contextual BERT embeddings consistently outperform FastText across all evaluated tasks. However, compressing non-contextual embeddings reduces their performance, making FastText more effective than compressed non-contextual BERT.

The key contributions of this work are as follows:

- We conduct a detailed study comparing non-contextual BERT embeddings and FastText embeddings for Marathi, a low-resource language. The evaluation covers multiple classification tasks, including sentiment analysis, news classification, and hate speech detection.

- To ensure a fair comparison, we compress BERT embeddings from 768 to 300 dimensions using Singular Value Decomposition (SVD). This allows us to analyze how dimensionality reduction impacts BERT's performance compared to its uncompressed version and FastText.

- We explore the differences between contextual and non-contextual BERT embeddings, examining their impact on classification performance in low-resource settings.

The paper is organized as follows: Section 2 provides a concise review of prior research on FastText and BERT. Section 3 includes the datasets and model embeddings that are utilized for the experiments. Section 4 presents the methodology used. Section 5 presents the results and key insights drawn from the findings along with a comparative analysis of FastText embeddings and BERT. In Section 6, we analyze our results and explain the reasons behind them. In Section 7, we conclude our discussion.

## 2 Literature Review

The existing literature emphasizes the superiority of contextual BERT embeddings over traditional word embedding techniques like Word2Vec (Mikolov et al., 2013) , GloVe (Pennington et al., 2014), and FastText across various natural language processing (NLP) tasks. For instance, Khaled et al. (2023) compare four popular pre-trained word embeddings—Word2Vec (via Aravec (Mohammad et al., 2017)), GloVe, FastText, and contextual BERT (via ARBERTv2)—on Arabic news datasets. They highlight BERT's superior performance, achieving over 95% accuracy due to its contextual interpretation.

Similarly, Kabullar and Türker (2022) analyzes the performance of embeddings on the AG News dataset, which includes 120K instances across four classes. They conclude that contextual BERT outperforms other methods, achieving 90.88% accuracy. FastText, Skip-Gram, CBOW, and GloVe achieve 86.91%, 85.82%, 86.15%, and 80.86%, respectively.

While traditional embeddings perform reasonably well, the consistent dominance of contextual BERT in complex tasks is also noted in sentiment analysis. For instance, Xie et al. (2024) explores how combining BERT and FastText embeddings enhances sentiment analysis in education, demonstrating that BERT's contextual understanding, along with FastText's ability to handle out-of-vocabulary words, improves generalization over unseen text.

In the domain of toxic speech classification, D'Sa et al. (2020) utilize both contextual BERT and FastText embeddings to classify toxic comments

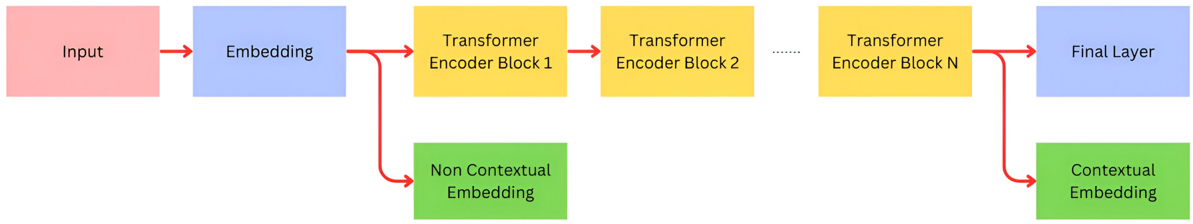



Figure 1: Embedding extraction workflow for contextual and non-contextual representations

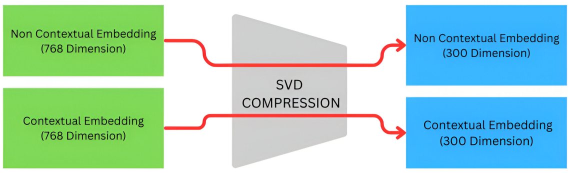



Figure 2: SVD compression of BERT embeddings

in English, with BERT embeddings outperforming FastText. This trend continues in hate speech detection, where Rajput et al. (2021) find that neural network classifiers using contextual BERT embeddings perform better than those with FastText embeddings alone, further supporting BERT's effectiveness.

Additionally, Chanda (2021) assess contextual BERT embeddings against traditional context-free methods (GloVe, Skip-Gram, and FastText) for disaster prediction, demonstrating BERT's superior performance in combination with traditional machine learning and deep learning methods.

For low-resource languages (LRLs), Ahmed et al. (2024) examine methods like traditional TF-IDF, contextual BERT, and FastText embeddings within a CNN-BiLSTM architecture for detecting depressive texts in Bangla. Their results show that BERT embeddings yield the highest F1 score (84%), indicating their dominance over other methods. This suggests that BERT's efficacy extends even to LRLs.

In medical applications, Khan et al. (2024) proposes integrating contextual BERT embeddings with SVM for prostate cancer prediction. By incorporating both numerical data and contextual information from clinical text, they achieve 95% accuracy, far outperforming the 86% accuracy achieved with numerical data alone.

Moreover, Malik et al. (2021) uses both contextual BERT and FastText embeddings to preprocess a dataset of conversations from Twitter and Facebook. Applying various machine learning and deep learning algorithms, they find that CNN yields the best results, further demonstrating BERT's capabilities.

Finally, while Asudani et al. (2023) offers a comprehensive analysis of traditional word embeddings alongside more advanced techniques like ELMo and contextual BERT, providing insight into commonly used datasets and models for benchmarking, Umer et al. (2022) highlights the versatility of FastText in various domains, despite BERT's consistently superior performance.

We note that the reviewed literature highlights the consistent superiority of BERT embeddings across various NLP tasks and domains. However, most existing studies focus mainly on contextual BERT embeddings, but not on non-contextual embeddings. Moreover, these studies predominantly address high-resource languages, leaving low-resource languages like Marathi largely unexplored. In particular, there is a lack of research assessing the effectiveness of non-contextual BERT embeddings for Marathi. Additionally, the impact of dimensionality leveling, i.e. the efficacy of BERT embedding compression, has not been explored.

## 3 Datasets and Models Used

In our research work, we used 3 Marathi datasets, MahaSent: A 3-class sentiment analysis dataset (Pingle et al., 2023), MahaHate: A 2-class as well as a 4-class hate classification dataset (Patil et al., 2022) and MahaNews is a news categorization dataset consisting of three sub-datasets, each with 12 classes: Short Headline Classification (SHC), Long Document Classification (LDC), and Long Paragraph Classification (LPC) (Mittal et al., 2023).

We used two types of embeddings in our experiments: FastText and BERT embeddings. For FastText, we utilized both IndicFT (Kakwani et al., 2020) and MahaFT (Joshi, 2022) embeddings. This was because both models included a Marathi corpus as part of their training data. MahaFT, in par-

| Type | Model | MahaSent | MahaHate | | MahaNews | | |
|---|---|---|---|---|---|---|---|
| | | 3-class | 4-class | 2-class | SHC | LDC | LPC |
| Contextual | MahaBERT | 82.27 | 66.8 | 85.57 | 89.83 | 93.87 | 87.78 |
| | MahaBERT (Compressed) | 82.89 | 66.15 | 84.37 | 89.61 | 93.53 | 87.82 |
| | Muril | 81.64 | 64.55 | 84.00 | 89.54 | 93.64 | 87.33 |
| | Muril (Compressed) | 81.91 | 63.2 | 83.36 | 88.38 | 93.48 | 87.45 |
| FastText | IndicFT | 76.4 | 58.25 | 80.13 | 85.57 | 92.15 | 79.19 |
| | MahaFT | 78.62 | 62.75 | 81.76 | 85.89 | 92.62 | 80.32 |
| Non-Contextual | MahaBERT | 77.56 | 66.5 | 82.64 | 86.45 | 91.69 | 81.76 |
| | MahaBERT (Compressed) | 76.31 | 63.9 | 81.57 | 83.85 | 91.25 | 80.08 |
| | Muril | 76.58 | 65.77 | 81.79 | 85.95 | 91.61 | 81.36 |
| | Muril (Compressed) | 75.16 | 63.25 | 81.44 | 82.72 | 90.39 | 79.00 |

Table 1: Performance of model embeddings on MahaSent, MahaHate, and MahaNews datasets using Multiple Logistic Regression. Key: SHC = Short Headline Classification, LPC = Long Paragraph Classification, LDC = Long Document Classification

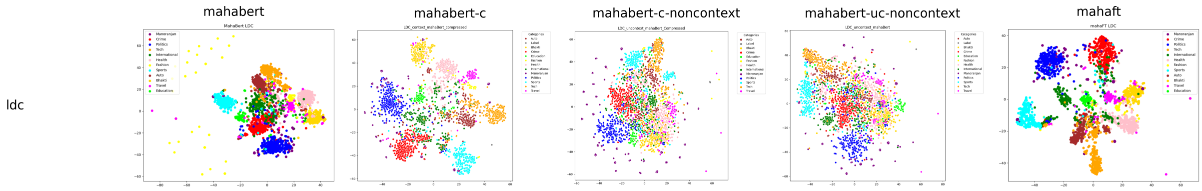



Figure 3: T-SNE Plot For BERT and FastText Embeddings (c stands for compressed) .

ticular, was specifically trained on a Marathi corpus, making it especially relevant for our experiments. For BERT embeddings, we primarily used two BERT-based models: MahaBERT (Joshi, 2022) and MuRIL (Khanuja et al., 2021). Since both models were trained on Marathi data, we selected them to compare with the FastText embeddings.

## 4 Methodology

For each sentence, corresponding embeddings were generated and the corresponding categorical labels were encoded into numerical labels. The creation of BERT embeddings was done by first tokenizing the text using the BERT tokenizer, along with padding and truncation. The tokenized input was then passed to the model and the output of the last hidden layer of BERT was taken, which was then averaged to get contextual embeddings for every sentence. Whereas for non-contextual embeddings, the output of the first embedding layer was used. Refer Figure 1 for the embedding extraction workflow for contextual and non-contextual representations.

However, for FastText, which is a non-contextual embedding by default, the process was slightly different due to the lack of a predefined vocabulary. Unlike BERT, which employs a tokenizer capable of processing entire Marathi sentences, FastText necessitates the creation of a custom vocabulary. To achieve this, the training and validation datasets were concatenated and passed through a text vectorizer, which generated vectors for every word in the dataset. The vectorizer returned the vocabulary as a list of words in decreasing order of their frequency. The FastText model was then loaded using the FastText library, and for each word in the vocabulary, a word vector was retrieved to construct the embedding matrix. For each sentence, the text was split into individual words, and the corresponding embeddings were retrieved from the embedding matrix. These embeddings were then averaged to produce the final sentence embeddings.

Additionally, we experimented with compressed embeddings by reducing the dimensionality from 768 (the traditional BERT embedding dimension) to 300. This compression was performed using Singular Value Decomposition (SVD) to select the most relevant features, extracting the top 300 components for all the combinations of contextual as well as non-contextual for MahaBERT as well as Muril. Refer Figure 2 for SVD compression of BERT embeddings.

Feature scaling was also applied to the outputs. All embeddings were then passed to a multiple logistic regression (MLR) classifier for classification into target labels.

| Dataset | Subdataset | Model | Avg | Variance | Std | Test |
|---|---|---|---|---|---|---|
| MahaSent | 3 Class | MahaBERT | 76.56 | 0.39843 | 0.6312 | 78.01 |
| | | MahaBERT-Compressed | 74.42 | 0.8498 | **0.9218** | 75.51 |
| | | Muril | 75.53 | 0.75268 | **0.8676** | 76.53 |
| | | Muril-Compressed | 72.97 | 0.48963 | 0.6997 | 75.2 |
| | | MahaFT | 77.28 | 0.38282 | 0.6187 | 78.58 |
| MahaHate | 4 Class | MahaBERT | 64.92 | 0.25203 | 0.5020 | 66.1 |
| | | MahaBERT-Compressed | 62.77 | 0.53875 | 0.7340 | 64.1 |
| | | Muril | 63.51 | 0.35307 | 0.5942 | 65.15 |
| | | Muril-Compressed | 61.22 | 0.52378 | 0.7237 | 62.9 |
| | | MahaFT | 62.48 | 0.22608 | 0.4755 | 62.55 |
| | 2 Class | MahaBERT | 84.23 | 0.37633 | 0.6135 | 82.53 |
| | | MahaBERT-Compressed | 82.3 | 0.10312 | 0.3211 | 81.41 |
| | | Muril | 83.69 | 0.39397 | 0.6277 | 81.63 |
| | | Muril-Compressed | 81.67 | 0.20943 | 0.4576 | 81.41 |
| | | MahaFT | 83.75 | 0.52153 | 0.7222 | 82.61 |
| MahaNews | SHC | MahaBERT | 86.66 | 0.27687 | 0.5262 | 86.64 |
| | | MahaBERT-Compressed | 84.13 | 0.36002 | 0.6000 | 83.81 |
| | | Muril | 85.7 | 0.06973 | 0.2641 | 85.66 |
| | | Muril-Compressed | 82.89 | 0.11612 | 0.3408 | 82.01 |
| | | MahaFT | 87.25 | 0.17873 | 0.4228 | 85.97 |
| | LDC | MahaBERT | 92.47 | 0.32565 | 0.5707 | 91.69 |
| | | MahaBERT-Compressed | 91.41 | 0.01637 | 0.1279 | 91.57 |
| | | Muril | 92.03 | 0.19055 | 0.4365 | 91.69 |
| | | Muril-Compressed | 91.04 | 0.07753 | 0.2784 | 90.39 |
| | | MahaFT | 92.79 | 0.15667 | 0.3958 | 92.71 |
| | LPC | MahaBERT | 81.71 | 0.18503 | 0.4302 | 81.27 |
| | | MahaBERT-Compressed | 80.03 | 0.1779 | 0.4218 | 80.51 |
| | | Muril | 81.19 | 0.17597 | 0.4195 | 81.4 |
| | | Muril-Compressed | 78.82 | 0.14497 | 0.3807 | 79.11 |
| | | MahaFT | 80.15 | 1.25257 | **1.1192** | 80.32 |

Table 2: The values were obtained by performing 5-fold cross-validation on the training dataset for **Non-contextual embedding**. The **Avg**, **Variance** and **Std** represent the average, variance and standard deviation respectively performance across the five test subsets (from training) of the 5-fold splits, while the **Test** column reflects the performance on the actual test dataset. Key: SHC = Short Headline Classification, LPC = Long Paragraph Classification, LDC = Long Document Classification

### 4.1 Experimental Setup

The experiments were conducted on Kaggle notebooks equipped with a P100 GPU accelerator, utilizing 16 GB of GPU memory, 20 GB of storage, and 32 GB of RAM. Accuracy was chosen as the evaluation metric, given the balanced nature of the datasets. For classification, the results obtained from the embeddings were mapped to final labels using a multinomial logistic regression model to maintain methodological simplicity. To determine the validity of the results obtained, 5-fold cross-validation was performed for all tasks, and the results are presented in Table 2.

### 4.2 Visualisation of Embeddings

To visualize how BERT and FastText embedding can separate the classes, we plotted T-SNE (van der Maaten and Hinton, 2008) graphs for the LDC dataset. We have 5 plots, with 4 plots for MahaBERT and 1 for MahaFT. Refer Figure 3.

## 5 Results

Table 1 presents the results for various embeddings, including MahaBERT, MuRIL, MahaFT, and IndicFT, across multiple datasets and tasks. It includes both contextual and non-contextual embeddings, as well as the compressed variants of MahaBERT and MuRIL.

In sections 5.1 and 5.2, we have considered the uncompressed versions of Muril and MahaBERT. Further, in section 5.3, we specifically show the effect of compression on Muril and MahaBERT.

### 5.1 Contextual vs FastText

From Table 1, we observe the following trend when comparing contextual embeddings with FastText embeddings: MahaBERT > MuRIL > MahaFT > IndicFT.

### 5.2 Non-Contextual vs FastText

The trend of comparing non-contextual embeddings with FastText typically follows this order:

MahaBERT > MuRIL > MahaFT > IndicFT. However, there are exceptions for the MahaSent and LDC datasets.

For these two datasets, FastText tends to perform slightly better. However, the difference is minimal, so we refer to Table 2 to determine whether this deviation is significant or simply random noise. We observe a high variance in MahaSent, suggesting that its deviation from the usual trend when comparing non-contextual embeddings with FastText may be attributed to noise and is unlikely to be significant.

In contrast, the LDC dataset also deviates from the trend but exhibits relatively low variance. As a result, for the LDC dataset, the performance trend when comparing non-contextual embeddings with FastText is as follows: MahaFT > IndicFT > MahaBERT > MuRIL.

### 5.3 Effect of Compression

From Table 1, it can be inferred that compression negatively impacts non-contextual embeddings, as uncompressed versions generally perform better. This is evident from MahaFT outperforming the compressed non-contextual MahaBERT embeddings in all datasets except MahaHate-4c, suggesting that compression lowers the performance of non-contextual BERT embeddings.

However, the effect of compression on contextual embeddings varies across datasets, making it challenging to derive a consistent conclusion.

## 6 Inference

In this section, we explain why the non-contextual MahaBERT embeddings outperform FastText (MahaFT and IndicFT) embeddings. Both MahaBERT and MahaFT embeddings have been trained on the same corpus of 752 million tokens Joshi (2022). The superior performance of non-contextual MahaBERT embeddings can be attributed to its larger embedding size, training data size, and contextual training objective. Specifically, the embedding size for Marathi-BERT-v2 is 152M (197,285 × 768), compared to MahaFT, which is 132M (439,247 × 300).

IndicFT performs worse than MahaFT, likely due to its smaller dataset size of 551 million tokens (Kakwani et al., 2020). On the other hand, contextual BERT achieves better results because its hidden layers are effectively utilized.

Additionally, we observe a negative impact when compressing MahaBERT non-contextual embeddings. Reducing the embedding size from 152M (197,285 × 768) to 59M (197,285 × 300) leads to a decrease in performance, likely due to the loss of representational capacity.

## 7 Conclusion

In our research, we analyzed the effectiveness of various BERT and FastText-based embeddings on three key NLP tasks for Marathi: news classification, hate speech classification, and sentiment classification focusing primarily on non-contextualised BERT embeddings.

Our results show that contextual BERT embeddings perform better than non-contextual ones, including both non-contextual BERT embeddings and FastText. Among non-contextual embeddings, BERT generally outperforms FastText in most tasks. However, when non-contextual BERT embeddings are compressed, their performance drops, and FastText performs better than compressed non-contextual BERT.

## References


Saad Ahmed, Mahdi H Sazan, Miraz A B M Muntasir, Rahman, Saad Ahmed Sazan, Mahdi H. Miraz, and M Muntasir Rahman. 2024. Enhancing depressive post detection in bangla: A comparative study of tf-idf, bert and fasttext embeddings. *ArXiv*, abs/2407.09187.

Deepak Suresh Asudani, Naresh Kumar Nagwani, and Pradeep Singh. 2023. Impact of word embedding models on text analytics in deep learning environment: a review. *Artificial Intelligence Review*, pages 1 – 81.

Ashis Kumar Chanda. 2021. Efficacy of bert embeddings on predicting disaster from twitter data. *ArXiv*, abs/2108.10698.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Ashwin Geet D'Sa, Irina Illina, and D. Fohr. 2020. Bert and fasttext embeddings for automatic detection of toxic speech. *2020 International Multi-Conference on: "Organization of Knowledge and Advanced Technologies" (OCTA)*, pages 1–5.

Raviraj Joshi. 2022. L3Cube-MahaCorpus and MahaBERT: Marathi monolingual corpus, Marathi BERT language models, and resources. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages

97–101, Marseille, France. European Language Resources Association.

Elif Kabullar and İlker Türker. 2022. Performance comparison of word embedding methods in text classification for various number of features.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

M Moneb Khaled, Muhammad Al-Barham, Osama Ahmad Alomari, and Ashraf Elnagar. 2023. Arabic news articles classification using different word embeddings. In *International Conference on Emerging Trends and Applications in Artificial Intelligence*, pages 125–136. Springer.

Asma Sadia Khan, Fariba Tasnia Khan, Tanjim Mahmud, Salman Karim Khan, Nahed Sharmen, Mohammad Shahadat Hossain, and Karl Andersson. 2024. Integrating bert embeddings with svm for prostate cancer prediction. In *2024 6th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, pages 01–06. IEEE.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

Pranav Malik, Aditi Aggrawal, and Dinesh Kumar Vishwakarma. 2021. Toxic speech detection using traditional machine learning models and bert and fasttext embedding with deep neural networks. *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pages 1254–1259.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Preprint*, arXiv:1301.3781.

Saloni Mittal, Vidula Magdum, Sharayu Hiwarkhedkar, Omkar Dhekane, and Raviraj Joshi. 2023. L3cube-mahanews: News-based short text and long document classification datasets in marathi. In *International Conference on Speech and Language Technologies for Low-resource Languages*, pages 52–63. Springer.

Abu Bakr Mohammad, Kareem Eissa, and Samhaa El-Beltagy. 2017. Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, 117:256–265.

Hrushikesh Patil, Abhishek Velankar, and Raviraj Joshi. 2022. L3cube-mahahate: A tweet-based marathi hate speech detection dataset and bert models. In *Proceedings of the Third Workshop on Threat, Aggression and Cyberbullying (TRAC 2022)*, pages 1–9.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Aabha Pingle, Aditya Vyawahare, Isha Joshi, Rahul Tangsali, and Raviraj Joshi. 2023. L3cube-mahasent-md: A multi-domain marathi sentiment analysis dataset and transformer models. *arXiv preprint arXiv:2306.13888*.

G. K. Rajput, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. 2021. Hate speech detection using static bert embeddings. *ArXiv*, abs/2106.15537.

Sello Ralethe. 2020. Adaptation of deep bidirectional transformers for Afrikaans language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2475–2478, Marseille, France. European Language Resources Association.

Muhammad Umer, Zainab Imtiaz, Muhammad Ahmad, Michele Nappi, Carlo Maria Medaglia, Gyu Sang Choi, and Arif Mehmood. 2022. Impact of convolutional neural network and fasttext embedding on text classification. *Multimedia Tools and Applications*, 82:5569–5585.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Pan Xie, Hengnian Gu, and Dongdai Zhou. 2024. Modeling sentiment analysis for educational texts by combining bert and fasttext. *2024 6th International Conference on Computer Science and Technologies in Education (CSTE)*, pages 195–199.

# Kantika: A Knowledge-Radiant Framework for Dermatology QA using IR-CoT and RAPTOR-Augmented Retrieval

**Deep Das and Vikram Singh and Dr. Rahul Dixit and Dr. Rohit Kumar**
Department of Artificial Intelligence
Sardar Vallabhbhai National Institute of Technology
Surat, India
u23ai052@coed.svnit.ac.in

## Abstract

This paper presents an improved Retrieval-Augmented Generation (RAG) approach for domain-specific question-answering in dermatology and cosmetic science. The proposed system integrates RAPTOR-style hierarchical indexing with Iterative Retrieval Chain-of-Thought (IR-CoT) reasoning and CRAG-style interleaved retrieval-generation to better handle complex, clinically grounded queries. It leverages multi-source dermatology data, including peer-reviewed research, product formulations, user reviews, and ingredient safety databases.

By decomposing queries into rationale-driven substeps and applying subgoal-specific retrieval, the system improves answer depth, accuracy, and relevance—particularly for ingredient interactions and personalized dermatological guidance. Empirical results show notable gains over standard RAG baselines in both precision and clinical coherence, establishing the effectiveness of this approach in specialized medical QA tasks. With 100% user satisfaction and 99.07% overall accuracy across all document categories, the system sets a strong benchmark for domain-specific medical QA in dermatology.

**Keywords — Retrieval Augmented Generation, IR-COT, CRAG, RAPTOR, Dermatology, Healthcare.**

## 1 Introduction

Dermatology, which is an integral part of the medical domain, presents unique challenges to question-answering systems due to the interaction of scientific knowledge, individual variations, and rapidly evolving product formulations. Users seeking dermatology advice require accurate and personalized information that considers multiple factors, including skin type, ingredient interactions, environmental conditions, and individual sensitivities.

With the increasing demand for personalized skincare and dermatological consultations, there is a growing need for AI systems that can deliver context-aware, medically grounded, and trustworthy responses. General-purpose models often fail to capture the granularity and layered reasoning required in this domain, making specialized solutions essential. Furthermore, most users seeking skincare advice are not medically trained, which means the answers must not only be accurate but also interpretable and reliable.

Traditional RAG systems often struggle with domain-specific queries that require multistep reasoning and integration of diverse information sources; also they lack when the question is of a broader context and its answer cannot be satisfactorily derived from a single knowledge source. Although general-purpose RAG architectures have shown success in broad knowledge domains, they face limitations when dealing with specialized domains that require hierarchical understanding and contextual reasoning. But the fact can't be denied that RAG has emerged as a powerful tool for knowledge-intensive natural language processing tasks (Lewis et al., 2020). Combining parametric knowledge from large language models with non-parametric knowledge retrieved from external sources has become a standard approach in recent RAG-related work. However, these approaches often struggle with complex reasoning tasks that require multi-step inference. Also, these are not reliable the medical domain where precision in the answering is of utmost importance.

This paper introduces an innovative RAG architecture specifically designed for answering dermatology domain questions. Our approach is combination of three steps which include RAPTOR-style hierarchical indexing that creates



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 34–41, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-005

reasoning-based document representations, Iterative Retrieval Chain-of-Thought (IR-CoT) that decomposes complex queries into manageable sub-questions, and CRAG-style interleaved retrieval and generation that maintains context throughout the reasoning process. Our main contributions are:

- A comprehensive RAG architecture tailored for domain-specific dermatology question answering
- Integration of RAPTOR indexing with IR-CoT retrieval to improve reasoning capabilities
- Thoroughly evaluated and market-ready deployable framework demonstrating superior performance in answer quality and faithfulness
- Open-source implementation enabling reproducibility and further research

## 2 Related Work

Recent works in RAG architectures have focused on improving retrieval quality and reasoning capabilities. Gao et al. (2023) introduced iterative retrieval mechanisms that refine queries based on intermediate results. Karpukhin et al. (2020) developed dense passage retrieval methods that better capture semantic similarity between queries and documents. Self-RAG (Asai et al., 2023) introduced self-reflection mechanisms that allow models to validate and improve their own outputs.

The RAPTOR framework (Sarthi et al., 2024) introduced tree-based indexing that creates hierarchical representations of document collections. Unlike traditional flat indexing approaches, RAPTOR constructs reasoning trees that capture both local and global document relationships. This approach is selected in our approach so that we can get more accurate data that is passed to the large language model. This is one of the important steps which can increase the reliability of answers in such crucial domains.

Chen et al. (2024) extended this concept with interactive reading mechanisms that dynamically navigate document hierarchies. These approaches demonstrate the importance of structured knowledge representation in retrieval systems.

Chain-of-Thought (CoT) prompting has demonstrated significant improvements in language model reasoning capabilities (Wei et al., 2022). The IR-CoT approach (Trivedi et al., 2022) extends this concept by interleaving the retrieval and reasoning steps, allowing for more dynamic and context-aware information gathering which is important for healthcare domain.

Yao et al. (2022) introduced ReAct, which combines reasoning and acting in language models, enabling more complex tool usage and multi-step problem solving. These approaches have proven particularly effective in complex question answering scenarios like dermatology.

Medical and healthcare domain question answering has received considerable attention due to the critical importance of accurate information (Shen et al., 2020). However, dermatology and cosmetics represent a unique subdomain with distinct challenges including ingredient interactions, individual variations, and rapidly evolving product formulations.

Zhang et al. (2023) developed early work on dermatological ingredient analysis using natural language processing, but focused primarily on ingredient classification rather than comprehensive question answering. Our work extends this by providing a complete RAG architecture for the domain. With Kantika, we try to provide a solution with real-world impact, we target an unexplored and existing problem which needs attention by providing a user-centric product which could be trusted and relied upon.

## 3 Proposed Methodology

Kantika represents a comprehensive RAG architecture specifically designed for answering dermatological questions that integrates multiple RAG techniques. The system addresses unique challenges of medical information processing where clinical accuracy, safety considerations, and comprehensive knowledge integration are paramount. Taking inspiration from the complex nature of dermatological practice, our proposed methodology combines hierarchical knowledge representation with iterative reasoning and adaptive generation to mirror the clinical approach of experienced dermatologists. As illustrated in Figure 1, our approach integrates retrieval-augmented prompting and causal reasoning through a multistage flow.

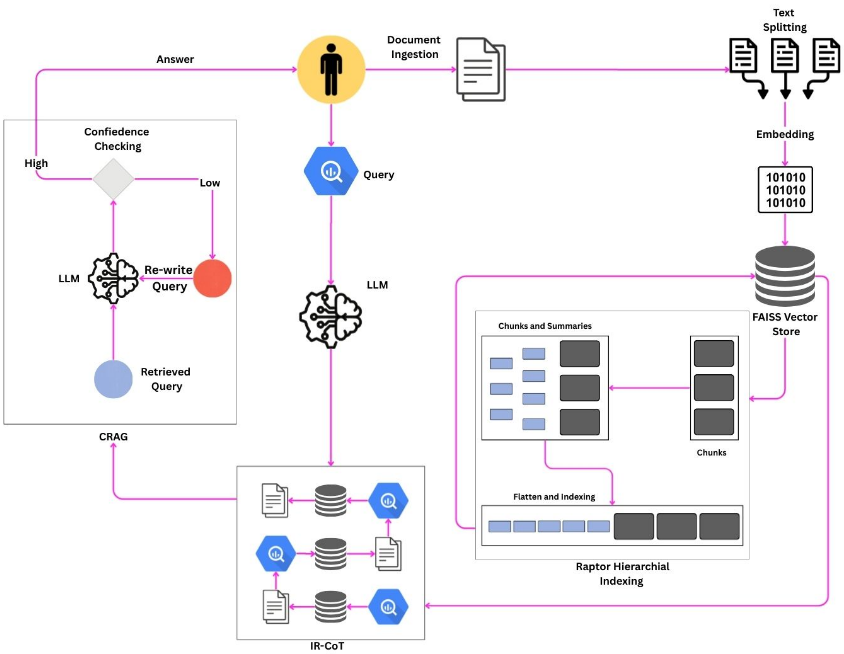



Figure 1: Flowchart of the proposed methodology combining RAPTOR, IR-CoT, and CRAG for dermatology-specific reasoning and retrieval.

### 3.1 Data Ingestion and Preprocessing

Kantika is built on a indigenously curated dataset that captures maximum possible scope of dermatological knowledge. The details of the dataset can be found in section 4.1. The multi-source setup we use tries to cover every possible aspect needed for great real-world outcomes and compliance standards. For preprocessing, we use *RecursiveCharacterTextSplitter* with 750 character chunks and 100 character overlap to keep preserve the context while maintaining semantic flow, which is crucial for medical accuracy.

### 3.2 RAPTOR-Style Hierarchical Indexing

The hierarchical indexing system transforms traditional flat document representations into structured reasoning pipeline that mimics the actual dermatological decision-making approach. Using our own specifically designed prompt templates, advanced language models extract 3-5 logical reasoning sub-questions from each document segment, creating a knowledge graph that thinks like a dermatologist. For example, when processing a document about retinol, the system builds a reasoning pipeline that capture molecular mechanisms, skin type compatibility, contraindications, and usage protocols. This pipeline forms a dynamic queryable structure that mimics a dermatologist's thinking process , evaluating multiple factors together for a user query. The knowledge graph captures both fine-grained document-level insights and broader clinical relationships, allowing it to provide context-aware evidence-backed recommendations for dermatological care.

### 3.3 Vector Storage and Semantic Embedding

The extracted reasoning nodes are embedded using Sentence Transformers models specifically fine-tuned for medical and scientific literature, ensuring precise semantic representation of clinical concepts. These embeddings are stored in FAISS vector databases for high-dimensional similarity search and enabling efficient retrieval of relevant information. The system implements the Maximum Marginal Relevance (MMR) search algorithm to balance semantic relevance with information diversity, ensuring that retrieved documents cover multiple aspects of a query rather than semantically similar but informationally redundant parts. This approach is particularly useful in dermatological applications where treatment recommendations must consider all aspects of efficacy, safety, interactions, and patient compliance factors.

### 3.4 IR-CoT Retrieval Strategy

The Iterative Retrieval Chain-of-Thought method breaks down complex dermatology queries into 2 to 4 logical sub-questions, similar to how a clinical practitioner evaluates a case. It follows the same layered reasoning used by experienced practitioners when dealing with multifactorial queries. For example, a query like combining vitamin C with niacinamide for hyperpigmentation gets split into mechanism analysis, interaction checks, best-use protocols, and skin-type-specific precautions. Each sub-question runs its own targeted retrieval step, pulling from relevant documents across ingredient science, clinical dermatology, and practical usage. This iterative structure allows the system to adapt to the complexity of the query and the clinical context involved.

### 3.5 CRAG-Style Interleaved Generation

The Corrective Retrieval-Augmented Generation (CRAG) workflow follows a consultation style model using a ReAct-based agent that can actively request more information when needed. If the system detects low confidence in the initial response based on predefined uncertainty thresholds, it triggers additional retrieval steps to ensure even complex or ambiguous queries are fully addressed. This adaptive setup reflects how real dermatology consultations vary in depth depending on the query, patient profile, and clinical context. CRAG preserves context over repeated reasoning steps allowing it to integrate information across multiple stages.

### 3.6 Answer Generation and Clinical Grounding

The final answer is generated using a medically tuned RetrievalQA system that grounds every response in relevant documents while strictly following evidence-based practices. The output is structured in a clinical format starting with the main recommendation backed by strong evidence, followed by the mechanism of action, precautions , user-specific considerations, and clear source citations. Every answer includes detailed source references allowing users can cross-check against the original medical literature. The answers are generated in simple english allowing any user to understand it. The system focuses on being clinically accurate, safe, and practically useful, while still being easy to understand for different types of users.

## 4 Experiments and Results

### 4.1 Knowledge Base Construction

The knowledge backend fed to the RAG system is a curated collection of authoritative dermatology textbooks sourced from leading academic and clinical publishers. This knowledge base consists of four works that together constitute the gold standard in dermatological clinical practices and education. The Oxford Handbook of Medical Dermatology, written by Susan Burge, Rubeta Matin, and Dinny Wallis is used as the master clinical reference framework ,

### 4.2 Evaluation Framework and Dataset Construction

The evaluation methodology employed in this research adheres to rigorous academic standards while incorporating both qualitative and quantitative assessment. Our comprehensive evaluation framework consists of two complementary approaches designed to assess system performance from multiple perspectives, ensuring robust validation suitable for medical AI applications.

#### 4.2.1 Human Evaluation

The human evaluation component involved 100 university students who were provided comprehensive access to the dermatology textbooks systematically integrated into Kantika's knowledge base. Participants were instructed to formulate questions directly from the inserted documentation, enabling the system to generate responses. They were then asked to evaluate their satisfaction with the system's answers, any instance of hallucination or misunderstanding, no matter how minor, was treated as a negative response to uphold our high standards. This methodology ensures that the evaluation questions are grounded in authoritative medical literature rather than arbitrary or potentially biased queries, thereby maintaining clinical relevance and educational validity.

#### 4.2.2 Automated Evaluation

The automated evaluation component follows established protocols in the retrieval-augmented generation research domain, utilizing the Mistral 7B Instruct model with 4-bit quantization to generate a comprehensive set of 430 evaluation questions. These are systematically categorized into three standard types that collectively assess different aspects of RAG system performance:

- **Single Document Queries:** Comprising 150 questions, these evaluate the system's ability to accurately retrieve and synthesize information from individual sources within the knowledge base.

- **Multi-Document Queries:** Totaling 200 questions, these assess the system's capacity for complex reasoning and cross-referencing capabilities across multiple authoritative sources.

- **Irrelevant Queries:** Consisting of 80 out-of-scope questions, these serve as a critical hallucination detection mechanism, ensuring that the system appropriately identifies questions whose answers are not present in the knowledge base and avoids generating fabricated medical information.

Each category serves a distinct purpose in validating system reliability, single document evaluation demonstrates precision in information retrieval and synthesis from individual sources, which is fundamental for answering specific clinical queries. Multi-document assessment evaluates the system's reasoning capabilities required for comprehensive clinical decision-making that often necessitates integrating information from multiple authoritative sources. Hallucination detection ensures clinical safety and trustability by validating the system's ability to recognize the boundaries of its knowledge and avoid generating potentially harmful unsubstantiated medical claims.

### 4.3 Results Analysis and Performance Assessment

The evaluation results demonstrate exceptional performance across both human and automated assessment protocols, establishing Kantika as a highly effective system for dermatological question answering. The human evaluation component yielded a remarkable 100% satisfaction rate across all 100 participating students, indicating unanimous approval of system responses when evaluated against questions formulated directly from authoritative medical textbooks. This exceptional satisfaction rate suggests that Kantika consistently provides clinically accurate, comprehensive, and practically applicable answers that meet the expectations of users with foundational medical knowledge.

The results of the automated evaluation strongly support adds to it achieving an overall accuracy of 99.07% across 430 generated questions. Single-document queries achieved a perfect 100% accuracy, reflecting Kantika's strong ability to precisely extract relevant information from individual texts. Multi-document queries scored 98.50% accuracy, demonstrating robust reasoning and effective cross-referencing across multiple sources—one of the most complex challenges in medical question answering. These results are summarized in Table 1.

In the hallucination detection task, designed using 80 irrelevant queries, Kantika achieved 98.75% accuracy, failing to reject only one instance. This near-perfect performance highlights its reliability in clinical environments, where generating safe and factual information is critical. These results affirm the strength of our integrated RAPTOR-style indexing, IR-CoT retrieval, and CRAG-based generation approach in creating a clinically trustworthy dermatological QA system.

## 5 Conclusion

Kantika demonstrates that clinical-grade medical AI can be developed with precision, reliability, and real-world impact. Achieving 100% user satisfaction and 99.07% accuracy on 430 expert-level dermatology questions, it creates a new standard for RAG systems for domain-specific domains. Its performance on multi-document generation, hallucination control, and single-document answering makes it deployment-ready for real-world applications.

With the integration of RAPTOR-style hierarchical indexing, IR-CoT retrieval, and CRAG-based reasoning, Kantika presents a workflow that replicates the thought processes of clinicians—open-ended, structured, and safe. The system effectively manages the intricacies of medical knowledge by giving priority to evidence, context, and clinical safety which are the three pillars vital to creating trustworthy AI in the healthcare sector.

Supported by peer-reviewed science and

| Question Category | Count | Performance | Description |
|---|---|---|---|
| Single Document Queries | 150 | 100% | Context derived from a single document in the knowledge base |
| Multi-Document Queries | 200 | 98.50% | Context requiring synthesis from multiple documents in the knowledge base |
| Irrelevant/Hallucination Detection | 80 | 98.75% | Domain-related questions not answerable from the knowledge base to test hallucination prevention |
| **Overall Performance** | **430** | **99.07%** | **Total evaluation across all standard RAG categories** |

Table 1: Comprehensive automated evaluation results demonstrating superior performance across standard RAG assessment categories.

validated in clinically sound trials, Kantika demonstrates that AI can fulfill the promise of contemporary medicine. It is not merely a system, it is driving scalable, domain-specific clinical support systems. In the years ahead, Kantika's architecture can drive next-gen AI for multimodal diagnosis, patient-specific treatment, and long-term clinical guidance—always putting better care above all with complete medical integrity.

## Implementation Details

This section provides detailed implementation information for reproducibility.

### System Architecture

The technical implementation of Kantika is built upon a robust foundation of state-of-the-art libraries and frameworks, ensuring both reliability and scalability for deployment in clinical environments. The system is implemented using Python 3.9+ as the primary development platform, leveraging LangChain v0.1.20 for comprehensive document processing and orchestration capabilities. The core language processing functionalities are powered by Gemini 2.0 Flash through the Google Generative AI API, providing advanced natural language understanding and generation capabilities specifically optimized for medical domain applications. Vector storage and similarity search operations are handled by FAISS v1.8.0, which offers high-performance indexing and retrieval capabilities essential for large-scale medical knowledge bases. Semantic embedding generation is accomplished through Sentence-Transformers v2.7.0, ensuring precise representation of medical concepts and terminology. Additional support for advanced model integration is provided through Transformers v4.35.0 and PyTorch v2.1.0, enabling flexible adaptation to emerging language models and specialized medical AI architectures.

### Hyperparameter Configuration

The implementation of Kantika employs carefully optimized hyperparameters that have been systematically tuned to achieve optimal performance in dermatological question answering tasks. The document processing pipeline utilizes a chunk size of 750 characters with an overlap of 100 characters, ensuring adequate context preservation while maintaining computational efficiency. The retrieval mechanism is configured to retrieve a maximum of 6 documents per query with a limit of 2 documents per individual query component, balancing comprehensiveness with processing speed. The generation component operates with a temperature setting of 0.1 to ensure consistent and reliable outputs while minimizing hallucination risks. The MMR retrieval system employs a top-k value of 5 with a lambda diversity parameter of 0.5, optimizing the balance between semantic relevance and information diversity. The iterative reasoning process is constrained to a maximum of 4 reasoning steps, ensuring thorough analysis while preventing excessive computational overhead.

## Practical Deployability and Open-Source Release

This work has been developed using standard libraries and follows best practices in software engineering to ensure reliability, reproducibility, and ease of integration. The pipeline is designed to be practically deployable, enabling dermatologists to utilize it for real-world applications such as personalized skincare recommendations, ingredient compatibility analysis, and patient-specific advice. By leveraging advanced retrieval and reasoning mechanisms, the system provides actionable insights that can be directly applied in clinical and advisory settings.

To promote transparency and further research, the complete implementation will be released as open-source software at https://github.com/THE-DEEPDAS/SkinCare-RAG.

## Ethics Statement

The development of Kantika has been guided by a commitment to ethical principles in artificial intelligence and healthcare. The system is designed to provide clinically accurate, evidence-based, and secure recommendations, thereby ensuring that it meets the highest standards of medical integrity. All the data sources used in developing the knowledge base are open to the public, peer-reviewed, and authentic medical literature, thereby ensuring openness and trustworthiness. The assessment framework has been designed to remove bias by using all human assessment questions to be based on credible dermatology textbooks, thereby ensuring clinical relevance and educational integrity.

It is important to note that Kantika is intended to be a decision support tool for clinicians and healthcare professionals and not to replace clinical judgment. The users are advised to consult the clinicians for personalized advice and treatment. The major use-case still lies in those particular parts of dermatology where the answer couldn't cause harm or side-effects to a particular user, this can be taken care of by adding only those books or knowledge sources which has information which cannot be of serious damage to the user. The advice offered by the system relies on the existing body of knowledge and is not influenced by patient histories or individual clinical situations, which remain in the jurisdiction of licensed medical professionals. Adhering to these ethical standards, Kantika is intended to supplement, not replace, the value added by human professional expertise in dermatologic care.

## Limitations

While Kantika is highly responsive in responding to dermatological questions, we must report some limitations so that an unbiased assessment of its functionalities is possible. First and foremost, the system is only text-based information drawn from credible dermatological resources without the inclusion of multimodal features like clinical images or patient-specific data. This limitation prevents it from being appropriate where patient-specific data or visual examination plays a prominent role. This particular thing was not added in this work due to ethical concerns but can be looked upon in future work. Second, although the knowledge base of the system is enormous, it is constrained by the sources that were intentionally added while developing it. Thus, it may not be the latest that has appeared in dermatology research or account for regional differences in clinical practice. Thus, periodic updating of the knowledge base is required to make it current and authentic.

Finally, Kantika's reliance on computational power, particularly for multi-step reasoning and retrieval operations, can be difficult to realize in low-resource environments. Future efforts will attempt to optimize the system for such environments, including reducing computational requirements and offline capability.

By recognizing these constraints, our objective is to deliver a clear evaluation of Kantika's present abilities, while also pinpointing opportunities for enhancement in the future.

## References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Howard Chen, Ramakanth Pasunuru, Jason Weston, and Asli Celikyilmaz. 2024. Walking down the memory maze: Beyond context limit through interactive reading. *arXiv preprint arXiv:2310.05029*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinniu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Proceedings of the*

*2020 Conference on Empirical Methods in Natural Language Processing*, pages 6769–6781.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. 2024. Raptor: Recursive abstractive processing for tree-organized retrieval. *arXiv preprint arXiv:2401.18059*.

Kaitao Shen, Di Jin, Xiaoxin Bao, Lifu Huang, Jian Ni, Haohan Zhu, Chunyuan Xiao, and 1 others. 2020. Medqa: A large-scale medical question answering dataset. *Applied Sciences*, 10(16):5421.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Wei Zhang, Mei Liu, Xiaojun Chen, and Qing Wang. 2023. Dermatological ingredient analysis and recommendation system using natural language processing. *Journal of Cosmetic Dermatology*, 22(4):1123–1135.

# GeistBERT: Breathing Life into German NLP

**Raphael Scheible-Schmitt[1,2] and Johann Frei[3]**

[1]School of Computation, Information and Technology, Technical University of Munich,
[2]IS[2]E - Intelligent Systems, Science and Engineering, LIACC polo on Azores University, Ponta Delgada, Portugal,
[3]Chair of IT Infrastructure for Translational Medical Research,
Faculty of Applied Computer Science, University of Augsburg

**Correspondence:** raphael.scheible@tum.de

## Abstract

Advances in transformer-based language models have highlighted the benefits of language-specific pre-training on high-quality corpora. In this context, German NLP stands to gain from updated architectures and modern datasets tailored to the linguistic characteristics of the German language. GeistBERT seeks to improve German language processing by incrementally training on a diverse corpus and optimizing model performance across various NLP tasks. We pre-trained GeistBERT using fairseq, following the RoBERTa base configuration with Whole Word Masking (WWM), and initialized from $\mathrm{GottBERT}$ weights. The model was trained on a 1.3 TB German corpus with dynamic masking and a fixed sequence length of 512 tokens. For evaluation, we fine-tuned the model on standard downstream tasks, including NER (CoNLL 2003, GermEval 2014), text classification (GermEval 2018 coarse/fine, 10kGNAD), and NLI (German XNLI), using $F_1$ score and accuracy as evaluation metrics. GeistBERT achieved strong results across all tasks, leading among base models and setting a new state-of-the-art (SOTA) in GermEval 2018 fine text classification. It also outperformed several larger models, particularly in classification benchmarks. To support research in German NLP, we release GeistBERT under the MIT license.

## 1 Introduction

The advancement of neural language modeling (LM) in natural language processing (NLP) has been driven by the development of contextual pre-trained word representations, particularly through transformer-based architectures. Models like Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) have significantly impacted the field by providing robust, generalized representations that can be fine-tuned for specific downstream tasks, enhancing performance across various NLP applications. While much of the early work focused on English and multilingual models, it has become clear that single-language models, particularly those trained on large, high-quality corpora, can outperform their multilingual counterparts when applied to their target language.

Building on this understanding, the German NLP community has seen the introduction of models like GottBERT (Scheible et al., 2024), which leveraged the German portion of the OSCAR (Ortiz Suárez et al., 2020) corpus to create a high-performance RoBERTa-based (Liu et al., 2019) model tailored specifically for the German language. However, as the field evolves, so too must the approaches to model training. Recent developments in pre-training methodologies, such as Whole Word Masking (WWM) (Cui et al., 2021) and the availability of newer, more extensive corpora like OSCAR23 (Jansen et al., 2022), OPUS (Tiedemann, 2012), and mC4 (Xue et al., 2021), present opportunities to further refine and enhance German language models.

To fully leverage these developments for German NLP, we introduce GeistBERT, a German Enhanced Incremental Semantically Tuned BERT model. GeistBERT builds on the foundation laid by the best checkpoint of the filtered GottBERT model (i.e. ${}^{\mathrm{f}}\mathrm{GottBERT}_{base}$) through continued pre-training (Gururangan et al., 2020a), extending it with modern German datasets including OSCAR23 and mC4 from CulturaX (Nguyen et al., 2023), Wikipedia, and several OPUS corpora. Since CulturaX already applies both deduplication and filtering, it provides a strong backbone of high-quality German text, while the additional corpora enrich the model with broader linguistic and domain diversity. By introducing Whole Word Masking (WWM) and leveraging the scale and variety of these sources, GeistBERT seeks to establish a new benchmark for German language models, with



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 42–50, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-006

improved performance across various NLP tasks.

Our contributions are as follows:

- We incrementally trained GeistBERT on top of $^{\mathrm{f}}\mathrm{GottBERT}_{base}$ using a combination of modern German corpora (OSCAR23, OPUS, mC4), OpenLegal and Wikipedia.

- We integrated WWM into the pre-training process to enhance the model's ability to capture semantic relationships within the German language.

- We provide GeistBERT as base model to the community, accessible under an open-source license for further usage.

GeistBERT represents a step forward in the development of German-specific transformer models, offering enhanced capabilities through modern training techniques and high-varying data.

## 2 Related Work

The rise of transformer-based models like BERT (Devlin et al., 2019) marked a major shift in NLP, enabling significant performance improvements. Originally introduced as an English model and later as a multilingual version (mBERT), BERT's success led to monolingual adaptations tailored to specific languages. For German, models like GermanBERT[1] and dbmdz BERT[2] emerged, trained on datasets of 12GB–16GB, sourced from Wikipedia, news articles, and legal texts.

RoBERTa enhanced BERT by training on a larger 160GB corpus, optimizing the architecture, and removing next sentence prediction. This strategy was applied to other languages, resulting in models like CamemBERT (Martin et al., 2020) for French and RobBERT (Delobelle et al., 2020) for Dutch, highlighting the benefits of large, diverse training corpora and the use of language-specific vocabularies.

In German NLP, GBERT and GELECTRA (Chan et al., 2020) built on this progress by training on 145GB of the OSCAR corpus (Ortiz Suárez et al., 2020) and additional sources, surpassing earlier German BERT models. These advancements underscored the impact of larger, well-curated datasets on model performance. GottBERT further extended this development as one of the first German RoBERTa models, trained on the German OSCAR corpus. Its results demonstrated the importance of data diversity but also noted that excessive data cleaning might reduce corpus variance and affect downstream performance. GeistBERT refines this lineage by increasing data variance, optimizing pre-training strategies, and achieving strong performance without increasing model size, making it a robust and accessible model for German NLP.

[1] https://www.deepset.ai/german-bert

[2] https://huggingface.co/dbmdz/bert-base-german-uncased

## 3 Methodology

### 3.1 Training Data and Pre-training

Compared to GottBERT, GeistBERT was trained on a substantially larger corpus, totaling approximately 1.3TB of text data. Training data was shuffled to support uniform sampling and minimize order effects during pre-training. GeistBERT was pre-trained using the same byte-level BPE tokenizer as GottBERT, following the GPT-2 design with a vocabulary size of 52k. While the tokenizer architecture mirrors GPT-2, the vocabulary itself was trained from scratch on German text. fairseq (Ott et al., 2019) was employed to compute the binary format for pre-training. Unlike GottBERT's TPU-based setup, which processed text as a continuous stream, GeistBERT's GPU training respected natural sentence boundaries. This preserves linguistic structure during pre-training and avoids cutting sequences in the middle of sentences.

Using fairseq, we pre-trained the GeistBERT model on a highly variant corpus consisting of 1.3TB plain text data on 8 NVIDIA A40 GPUs. The model was trained with the RoBERTa base architecture for 100k update steps using a batch size of 8k, initializing the weights with $^{\mathrm{f}}\mathrm{GottBERT}_{base}$. We largely adhered to RoBERTa's default training configuration (Liu et al., 2019), including dynamic masking, optimizer settings, and fixed sequence lengths (512 tokens). A 10k iteration warmup was applied, gradually increasing the learning rate to a peak of 0.0007, followed by a polynomial decay to zero.

### 3.2 Downstream Tasks

We fine-tuned pre-trained BERT models using Huggingface (Wolf et al., 2019) scripts, optimizing batch size and learning rate via grid search. NER and classification (CLS) tasks were trained for up to 30 epochs, while NLI tasks ran for up to 10 epochs using fairseq-adapted hyperparameters. Each task was executed 24 times with varied hyperparam-

Table 1: Overview of datasets used for training. The table lists the individual corpora, their sizes in gigabytes, their data sources, and whether they were deduplicated or filtered. The final corpus aggregates all listed datasets, resulting in approximately 1.3 TB of training data.

| Corpus | Documents | Size (GB) | Data Source | Deduplicated | Filtered |
|---|---|---|---|---|---|
| mC4 & OSCAR23 | 6,064,736,930 | 1316.57 | CulturaX | Yes | Yes |
| ELRC-4244, ELRC-4240, ELRC-4258, ELRC-4217, ELRC-4189, ELRC-4171, ELRC-4149 | 14,919,003 | 2.34 | OPUS | Yes | No |
| ECB | 1,732,472 | 0.29 | OPUS | No | No |
| EUbookshop | 18,203,612 | 2.34 | OPUS | No | No |
| Europarl | 2,234,583 | 0.36 | OPUS | No | No |
| EuroPat | 19,387,517 | 3.52 | OPUS | No | No |
| OpenSubtitles | 41,612,280 | 1.35 | OPUS | No | No |
| TildeMODEL | 5,059,688 | 0.79 | OPUS | No | No |
| German Wikipedia | 4,767,776 | 7.23 | Wikipedia | No | No |
| OpenLegalData | 209,526 | 2.48 | OpenLegal | No | No |
| **Final corpus** | **6,172,863,387** | **1337.28** | | | |

eters, selecting the best checkpoint based on the highest $F_1$ score (accuracy for NLI). Performance was evaluated analogously to Scheible et al. (2024) and compared with results from that study. The parameter search space used for the grid search is summarized in Table 2. All tasks were processed using two Nvidia RTX 3090 GPUs, leveraging Huggingface's Transformers library (v4.34.1).

Table 2: Hyperparameters used in the grid search of the downstream tasks.

| Parameter | Values |
|---|---|
| Learning Rate | 5e-5, 2e-5, 1e-5, 7e-6, 5e-6, 1e-6 |
| Batch Size | 16, 32, 48, 64 |
| Epochs | 30 |

**NLI** We evaluated NLI on the German XNLI dataset (Conneau et al., 2018), an extension of MultiNLI (Williams et al., 2018), with 122k training, 2490 development, and 5010 test examples per language. Performance was measured by accuracy.

**Named Entity Recognition** NER evaluation used the German CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) and GermEval 2014 (Benikova et al., 2014) datasets. CoNLL 2003 includes four entity types, while GermEval 2014 provides fine-grained categories and supports nested annotations. Both were evaluated using the $F_1$ score, with GermEval using an adapted metric accounting for label and span equality.

**Text Classification** We evaluated classification on GermEval 2018 (Risch et al., 2018) (German tweet sentiment analysis) and 10kGNAD (Schabus et al., 2017) (German news categorization). GermEval 2018 followed the data splits defined by Chan et al. (2020), while 10kGNAD used a predefined 90%-10% train–test split, with 10% of the training set further held out for validation. Both tasks were evaluated using the mean $F_1$ score.

### 3.3 Model Properties

Table 3 lists the vocabulary sizes and total parameter counts of all models included in our evaluation. While most German BERT-style base models, such as $\text{GBERT}_{\text{base}}$, dbmdzBERT, and $\text{GELECTRA}_{\text{base}}$, contain approximately 110 million parameters, GeistBERT and $^{\mathrm{f}}\mathrm{GottBERT}_{base}$ are slightly larger at around 126 million parameters due to their RoBERTa-based architecture and a larger vocabulary of 52,009 tokens.

Large-scale German models such as $\text{GBERT}_{\text{large}}$, $\text{GELECTRA}_{\text{large}}$, and $^{\mathrm{f}}\mathrm{GottBERT}_{large}$ contain between 335 and 357 million parameters. Among the multilingual models, $\text{XLM-RoBERTa}_{\text{base}}$ and $\text{XLM-RoBERTa}_{\text{large}}$ are substantially larger, with 278 million and 560 million parameters respectively. The vocabulary sizes vary across models and are influenced by tokenizer design and pretraining data. GeistBERT uses the same tokenizer as GottBERT, which is based on byte-level BPE trained on German text.

## 4 Results

### 4.1 Training Dynamics

During the model pre-training the perplexity of the model is computed based on a test set for each optimization cycle (see Figure 1). After an initial sharp decrease, perplexity briefly increased for several steps before gradually declining until the final step. We assume that, given more training time, it would have continued to decrease further. The entire pre-

Table 3: The size of the vocabulary and the size of the parameters are shown for the model types used in this study. This table does not show other design differences of the models. Values were extracted using Huggingface's transformers library.

| Model | Vocab Size | #Params |
|---|---|---|
| XLM-R$_{\text{large}}$ | 250002 | 559890432 |
| $^{\text{f}}\texttt{GottBERT}_{large}$ | 52009 | 357145600 |
| GBERT$_{\text{large}}$ | 31102 | 335735808 |
| GELECTRA$_{\text{large}}$ | 31102 | 334686208 |
| XLM-R$_{\text{base}}$ | 250002 | 278043648 |
| mBERT | 119547 | 177853440 |
| GeistBERT | 52009 | 125985024 |
| $^{\text{f}}\texttt{GottBERT}_{base}$ | 52009 | 125985024 |
| GBERT$_{\text{base}}$ | 31102 | 109927680 |
| dbmdzBERT | 31102 | 109927680 |
| GELECTRA$_{\text{base}}$ | 31102 | 109337088 |
| GermanBERT | 30000 | 109081344 |

training process required approximately 8.3 days of computation time.

Importantly, GeistBERT started from a relatively low perplexity due to continued pre-training. In comparison, $^{\text{f}}\texttt{GottBERT}_{base}$ (trained entirely from scratch) started with a perplexity of about 52,592 and converged to around 4, whereas GeistBERT began at 35.17 and converged down to approximately 11. This illustrates the potential stability and efficiency benefits of continued pre-training in reaching useful representations quickly.

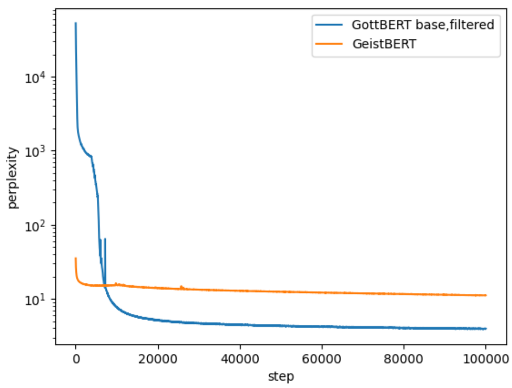



Figure 1: Perplexity of $^{\text{f}}\texttt{GottBERT}_{base}$ and GeistBERT, evaluated on the validation set after each optimization cycle; values are plotted on a logarithmic y-axis.

### 4.2 Downstream Tasks

GeistBERT sets a new state-of-the-art among base models for German NLP, outperforming all comparable models and closely approaching large-scale model performance across tasks. It even achieves absolute SOTA in GermEval 2018 fine-grained classification (see Table 6).

The optimal hyperparameters selected per task are summarized in Table 5, extending the original GottBERT setup (Scheible et al., 2024) by including GeistBERT models. The total computation time for all downstream evaluations amounted to 517 hours and 24 minutes (≈21.6 days) on two Nvidia RTX 3090 GPUs, detailed per task in Table 4.

Table 4: Computation time in hours and minutes for the downstream tasks summing up to 517 hours and 24 minutes, which are approximately 21.6 days.

| Task | | Computation Time |
|---|---|---|
| XNLI | | 47:52 |
| GermEval 2014 | | 235:36 |
| CoNLL03 | | 92:45 |
| GermEval 2018 | coarse | 45:46 |
| | fine | 43:25 |
| 10kGNAD | | 85:49 |

**NLI** GeistBERT$_{\text{base}}$ achieves an accuracy of 82.67% on the German NLI task, outperforming all other base models in our evaluation. While it does not surpass top-scoring large-scale models such as GELECTRA$_{\text{large}}$ (86.33%) or GBERT$_{\text{large}}$ (84.21%), it performs competitively and even surpasses $\texttt{GottBERT}_{large}$ (82.46%) and nearly matches $^{\text{f}}\texttt{GottBERT}^{\dagger}_{large}$ (82.79%), narrowing the performance gap despite its smaller size.

**Named Entity Recognition** GeistBERT achieves strong $F_1$ scores on both CoNLL 2003 (86.17%) and GermEval 2014 (88.47%), outperforming all other base models in our evaluation. It also surpasses all large-scale GottBERT variants on GermEval 2014 and comes remarkably close on CoNLL 2003, with only a 0.11% gap to the lowest-scoring large variant. While top-performing large models such as GBERT$_{\text{large}}$ (87.19% on CoNLL) and XLM-R$_{\text{large}}$ (88.83% on GermEval) remain ahead, GeistBERT narrows the performance gap significantly, demonstrating robust entity representation capabilities despite its compact size.

**Text Classification** GeistBERT$_{\text{base}}$ achieves strong performance across all classification tasks, ranking first in GermEval 2018 fine-grained classification (66.42%), second in 10kGNAD (90.89%), and third in GermEval 2018 coarse (79.67%). It consistently outperforms all other base models and surpasses several large-scale models, particularly

Table 5: Hyperparameters of the best downstream task models for each task and pre-trained model. This table extends the original GottBERT setup by including GeistBERT models. BS refers to batch size, and LR denotes the learning rate.

| **Model** | **GermEval 2014** | | **CoNLL 03** | | **GermEval 2018** coarse | | **GermEval 2018** fine | | **10kGNAD** | |
|---|---|---|---|---|---|---|---|---|---|---|
| | BS | LR | BS | LR | BS | LR | BS | LR | BS | LR |
| GeistBERT | 16 | 5 E-06 | 32 | 2 E-05 | 48 | 5 E-05 | 32 | 2 E-05 | 16 | 1 E-06 |
| $\mathrm{GottBERT}_{base}$ | 16 | 1 E-05 | 32 | 2 E-05 | 48 | 7 E-06 | 32 | 5 E-06 | 32 | 5 E-06 |
| $\mathrm{GottBERT}^{\dagger}_{base}$ | 48 | 2 E-05 | 32 | 5 E-05 | 48 | 1 E-05 | 64 | 7 E-06 | 32 | 5 E-06 |
| $^{f}\mathrm{GottBERT}_{base}$ | 16 | 7 E-06 | 16 | 1 E-05 | 16 | 1 E-05 | 48 | 2 E-05 | 16 | 5 E-06 |
| $^{f}\mathrm{GottBERT}^{\dagger}_{base}$ | 16 | 1 E-05 | 64 | 5 E-05 | 16 | 1 E-05 | 16 | 2 E-05 | 16 | 1 E-05 |
| $\mathrm{GELECTRA}_{\mathrm{base}}$ | 32 | 5 E-05 | 64 | 5 E-05 | 16 | 2 E-05 | 48 | 5 E-05 | 48 | 5 E-05 |
| $\mathrm{GBERT}_{\mathrm{base}}$ | 16 | 2 E-05 | 64 | 2 E-05 | 32 | 1 E-05 | 16 | 5 E-05 | 16 | 2 E-05 |
| dbmdzBERT | 48 | 2 E-05 | 48 | 5 E-05 | 16 | 5 E-06 | 64 | 2 E-05 | 16 | 2 E-05 |
| GermanBERT | 32 | 2 E-05 | 16 | 1 E-05 | 16 | 1 E-05 | 32 | 1 E-05 | 32 | 5 E-05 |
| $\mathrm{XLM\text{-}R}_{\mathrm{base}}$ | 64 | 2 E-05 | 16 | 1 E-05 | 48 | 5 E-05 | 64 | 5 E-05 | 48 | 2 E-05 |
| mBERT | 48 | 1 E-05 | 16 | 2 E-05 | 16 | 2 E-05 | 64 | 5 E-05 | 64 | 2 E-05 |
| $\mathrm{GottBERT}_{large}$ | 64 | 5 E-06 | 16 | 5 E-06 | 64 | 5 E-06 | 32 | 7 E-06 | 64 | 1 E-06 |
| $^{f}\mathrm{GottBERT}_{large}$ | 32 | 5 E-06 | 48 | 2 E-05 | 32 | 5 E-06 | 32 | 7 E-06 | 16 | 5 E-06 |
| $^{f}\mathrm{GottBERT}^{\dagger}_{large}$ | 16 | 5 E-06 | 48 | 1 E-05 | 48 | 1 E-05 | 32 | 5 E-06 | 64 | 2 E-05 |
| $\mathrm{GELECTRA}_{\mathrm{large}}$ | 16 | 7 E-06 | 16 | 5 E-06 | 64 | 1 E-05 | 32 | 2 E-05 | 32 | 2 E-05 |
| $\mathrm{GBERT}_{\mathrm{large}}$ | 16 | 7 E-06 | 32 | 5 E-06 | 16 | 2 E-05 | 64 | 2 E-05 | 64 | 5 E-05 |
| $\mathrm{XLM\text{-}R}_{\mathrm{large}}$ | 16 | 7 E-06 | 48 | 1 E-05 | 32 | 1 E-05 | 32 | 1 E-05 | 16 | 5 E-06 |

Table 6: All the results of the experiments are shown in percent. They are all based on the test set and the best score out of 24 runs (selection based on validation set). While NLI is measured by accuracy, all the other metrics are $F_1$ measures. Per model size, best results are **bold**, second-best underlined. Results for GottBERT are reported on both the unfiltered and filtered corpora, the latter indicated by $^{f}$. For each GottBERT model, we include both the best and last checkpoint of the pre-training, with the last denoted by $^{\dagger}$. Values for non-GeistBERT models are taken from Scheible et al. (2024).

| **Model** | **XNLI** | **GermEval 2014** | **CoNLL 03** | **GermEval 2018** coarse | **GermEval 2018** fine | **10kGNAD** |
|---|---|---|---|---|---|---|
| GeistBERT | **82.67** | **88.47** | **86.17** | **79.67** | **66.42** | **90.89** |
| $\mathrm{GottBERT}_{base}$ | 80.82 | 87.55 | 85.93 | 78.17 | 53.30 | 89.64 |
| $\mathrm{GottBERT}^{\dagger}_{base}$ | 81.04 | 87.48 | 85.61 | 78.18 | 53.92 | 90.27 |
| $^{f}\mathrm{GottBERT}_{base}$ | 80.56 | 87.57 | 86.14 | 78.65 | 52.82 | 89.79 |
| $^{f}\mathrm{GottBERT}^{\dagger}_{base}$ | 80.74 | 87.59 | 85.66 | 78.08 | 52.39 | 89.92 |
| $\mathrm{GELECTRA}_{\mathrm{base}}$ | 81.70 | 86.91 | 85.37 | 77.26 | 50.07 | 89.02 |
| $\mathrm{GBERT}_{\mathrm{base}}$ | 80.06 | 87.24 | 85.16 | 77.37 | 51.51 | 90.30 |
| dbmdzBERT | 68.12 | 86.82 | 85.15 | 77.46 | 52.07 | <u>90.34</u> |
| GermanBERT | 78.16 | 86.53 | 83.87 | 74.81 | 47.78 | 90.18 |
| $\mathrm{XLM\text{-}R}_{\mathrm{base}}$ | 79.76 | 86.14 | 84.46 | 77.13 | 50.54 | 89.81 |
| mBERT | 77.03 | 86.67 | 83.18 | 73.54 | 48.32 | 88.90 |
| $\mathrm{GottBERT}_{large}$ | 82.46 | 88.20 | <u>86.78</u> | 79.40 | 54.61 | 90.24 |
| $^{f}\mathrm{GottBERT}_{large}$ | 83.31 | 88.13 | 86.30 | 79.32 | 54.70 | 90.31 |
| $^{f}\mathrm{GottBERT}^{\dagger}_{large}$ | 82.79 | 88.27 | 86.28 | 78.96 | 54.72 | 90.17 |
| $\mathrm{GELECTRA}_{\mathrm{large}}$ | **86.33** | <u>88.72</u> | <u>86.78</u> | **81.28** | <u>56.17</u> | **90.97** |
| $\mathrm{GBERT}_{\mathrm{large}}$ | <u>84.21</u> | <u>88.72</u> | **87.19** | <u>80.84</u> | **57.37** | <u>90.74</u> |
| $\mathrm{XLM\text{-}R}_{\mathrm{large}}$ | 84.07 | **88.83** | 86.54 | 79.05 | 55.06 | 90.17 |

in the fine-grained setting. The results indicate that GeistBERT performs competitively across diverse classification benchmarks, despite being a base-sized model.

## 5 Discussion

### 5.1 Principal Findings

The continued pre-training of GottBERT on a broader and partially deduplicated and filtered German corpus consisting of OSCAR23, OPUS, mC4, Wikipedia, and OpenLegal, together with the use of WWM, leads to clear improvements across multiple language modeling tasks. GeistBERT establishes a new state of the art among base models and achieves competitive results with larger models across multiple German NLP benchmarks.

### 5.2 Training Considerations and Data Quality

In contrast to the TPU-based training used for GottBERT, GPU training also enabled more flexible preprocessing, such as sentence-aware segmentation. This made it possible to preserve natural sentence structure during training, even when using fixed-length sequences. Nevertheless, hyperparameter tuning remains a crucial factor for achieving strong downstream performance (Dodge et al., 2020). WWM contributed to improved tokenization, aligning with previous findings (Martin et al., 2020; Chan et al., 2020). However, we did not perform a dedicated ablation study comparing WWM with standard subword masking, as this would have required training an additional baseline model. Nevertheless, the consistently strong downstream results of GeistBERT suggest that WWM contributed positively, in line with earlier findings. Moreover, we were able to adopt a higher peak learning rate (0.0007), which may also have been facilitated by initializing from the $^{\mathrm{f}}\mathrm{GottBERT}_{base}$ checkpoint.

While deduplication and filtering were applied to CulturaX, other subcorpora (e.g., OPUS, Wikipedia, OpenLegal) were only partially processed or left unfiltered. This means that some redundant or lower-quality data may still be present. Prior work suggests that models benefit from increased corpus diversity (Martin et al., 2020), and GeistBERT's use of many different corpora likely contributed to its robustness. Additionally, vocabulary size plays a role in performance (Toraman et al., 2023), though ours remains well-optimized.

We did not perform ablation experiments per subcorpus, as this would have required multiple additional large-scale pre-training runs. Nevertheless, we expect that improvements are not only attributable to the sheer size of the training data (1.3 TB), but also to the increased heterogeneity of the sources. The OSCAR23+mC4 portion clearly contributed the majority of the volume, while smaller corpora such as OpenLegal, Wikipedia, and OPUS are likely to have increased linguistic and domain diversity. Prior findings from CamemBERT (Martin et al., 2020) indicate that variance of a corpus matter and impacts downstream robustness, which suggests that the mix of sources in GeistBERT was similarly beneficial.

### 5.3 Continued Pre-training and Outlook

We chose to continue pre-training from $\mathrm{GottBERT}$ rather than training GeistBERT from scratch, as it is common practice with domain-specific adaptations (Lentzen et al., 2022; Lee et al., 2019; Arefeva and Egger, 2022; Gururangan et al., 2020b). This allowed us to reuse German-specific tokenization and pre-trained weights, and to focus on training and evaluating a single, well-defined setup within time constraints. While training from scratch with a custom vocabulary may yield more tailored embeddings (El Boukkouri et al., 2022), prior work suggests that continued pre-training often achieves comparable results. A direct comparison between continued pre-training and training from scratch on the same architecture and corpus remains an interesting avenue for future work.

Following the broad adoption of GottBERT in German NLP (Scherrmann, 2023; Bressem et al., 2024; Lentzen et al., 2022; Xu et al., 2021; Frei et al., 2022; Frei and Kramer, 2023), we hope GeistBERT will be similarly received and applied.

## 6 Conclusion

In this work, we introduced GeistBERT, a German RoBERTa-based language model trained on a diverse as well as partially deduplicated and filtered corpus, incorporating WWM to enhance pre-training. GeistBERT achieves SOTA performance among base models and even outperforms several larger models across multiple tasks. These results underscore the importance of corpus diversity and WWM in improving downstream performance. GeistBERT is released under the MIT license on Huggingface, with fairseq checkpoints provided.

## Limitations

Several limitations should be acknowledged in this study. First, while deduplication and filtering were applied to CulturaX (OSCAR23 + mC4) and deduplication to selected OPUS corpora, other parts of the dataset (e.g., Wikipedia, OpenLegal) were not processed, potentially leaving redundant or noisy data.

Second, GeistBERT's training data, though diverse, remains specific to the selected corpora (OSCAR23, OPUS, mC4, Wikipedia, OpenLegal). Its generalization to other datasets or domains remains uncertain, and performance on dialects and cultural nuances within German may be limited. Further fine-tuning could improve adaptability to regional language variations.

Third, we did not include a detailed error analysis of model predictions. While such an analysis could provide additional insights into systematic failure modes, our focus in this work was on efficiency and establishing strong baselines for German NLP.

Finally, due to efficiency constraints and limited computational resources, we did not train a large version of GeistBERT, as pretraining based on GottBERT estimates would have required approximately 4.75 times more compute. While our results demonstrate the strong performance of the base model, larger architectures could potentially achieve even better results.

## Ethical Considerations

Like all large-scale language models, GeistBERT may inherit biases from its training data, which can influence downstream tasks such as classification or decision-making. While deduplication reduces redundancy and noise, it does not remove deeper societal or representational biases. Furthermore, training on large web-based corpora raises privacy concerns, as models may inadvertently retain sensitive information. Responsible deployment is especially important in high-stakes domains like legal, medical, or financial NLP.

Despite optimizations for efficiency, pre-training and evaluating transformer models remain computationally demanding, contributing to energy use and carbon emissions. These environmental costs highlight the need for balancing model performance with sustainable development goals.

## References


Veronika Arefeva and Roman Egger. 2022. When bert started traveling: Tourbert—a natural language processing model for the travel industry. *Digital*, 2(4):546–559.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Padó. 2014. GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, pages 104–112.

Keno K. Bressem, Jens-Michalis Papaioannou, Paul Grundmann, Florian Borchert, Lisa C. Adams, Leonhard Liu, Felix Busch, Lina Xu, Jan P. Loyen, Stefan M. Niehues, Moritz Augustin, Lennart Grosser, Marcus R. Makowski, Hugo J.W.L. Aerts, and Alexander Löser. 2024. medbert.de: A comprehensive german bert model for the medical domain. *Expert Systems with Applications*, 237:121598.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 29:3504–3514.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. 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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv:2002.06305 [cs]*. ArXiv: 2002.06305.

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, and Pierre Zweigenbaum. 2022. Re-train or train from scratch? comparing pre-training strategies of BERT in the medical domain. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2626–2633, Marseille, France. European Language Resources Association.

Johann Frei, Ludwig Frei-Stuber, and Frank Kramer. 2022. Gernermed++: Transfer learning in german medical nlp.

Johann Frei and Frank Kramer. 2023. Annotated dataset creation through large language models for non-english medical nlp. *Journal of Biomedical Informatics*, 145:104478.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020a. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020b. Don't stop pretraining: Adapt language models to domains and tasks. *CoRR*, abs/2004.10964.

Tim Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. 2022. Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Manuel Lentzen, Sumit Madan, Vanessa Lage-Rupprecht, Lisa Kühnel, Juliane Fluck, Marc Jacobs, Mirja Mittermaier, Martin Witzenrath, Peter Brunecker, Martin Hofmann-Apitius, Joachim Weber, and Holger Fröhlich. 2022. Critical assessment of transformer-based AI models for German clinical notes. *JAMIA Open*, 5(4):ooac087.

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:1907.11692 [cs]*. ArXiv: 1907.11692.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de la Clergerie, Djamé Seddah, and Benoît Sagot. 2020. CamemBERT: a Tasty French Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, Online. Association for Computational Linguistics.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. *arXiv:1904.01038 [cs]*. ArXiv: 1904.01038.

Julian Risch, Eva Krebs, Alexander Löser, Alexander Riese, and Ralf Krestel. 2018. Fine-Grained Classification of Offensive Language. In *Proceedings of GermEval 2018 (co-located with KONVENS)*, pages 38–44.

Dietmar Schabus, Marcin Skowron, and Martin Trapp. 2017. One Million Posts: A Data Set of German Online Discussions. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 1241–1244, Tokyo, Japan.

Raphael Scheible, Johann Frei, Fabian Thomczyk, Henry He, Patric Tippmann, Jochen Knaus, Victor Jaravine, Frank Kramer, and Martin Boeker. 2024. GottBERT: a pure German language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21237–21250, Miami, Florida, USA. Association for Computational Linguistics.

Moritz Scherrmann. 2023. German finbert: A german pre-trained language model.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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 - Volume 4*, CONLL '03, pages 142–147, USA. Association for Computational Linguistics. Event-place: Edmonton, Canada.

Cagri Toraman, Eyup Halit Yilmaz, Şahinuç Furkan, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Haoran Xu, Benjamin Van Durme, and Kenton Murray. 2021. Bert, mbert, or bibert? a study on contextualized embeddings for neural machine translation.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

# Identifying Contextual Triggers in Hate Speech Texts Using Explainable Large Language Models

Dheeraj Kodati[1] Lakkireddy Bhuvana Sree[1]

dheeraj.kodati@mahindrauniversity.edu.in se24maid009@mahindrauniversity.edu.in

[1]Department of Computer Science and Engineering, Mahindra University, Hyderabad, India

## Abstract

The pervasive spread of hate speech on online platforms poses a significant threat to social harmony, necessitating not only high-performing classifiers but also models capable of transparent, fine-grained interpretability. Existing methods often neglect the identification of influential contextual words that drive hate speech classification, limiting their reliability in high-stakes applications. To address this, we propose LLM-BiMACNet (Large Language Model-based Bidirectional Multi-Channel Attention Classification Network), an explainability-focused architecture that leverages pretrained language models and supervised attention to highlight key lexical indicators of hateful and offensive intent. Trained and evaluated on the HateXplain benchmark—comprising class labels, target community annotations, and human-labeled rationales—LLM-BiMACNet is optimized to simultaneously enhance both predictive performance and rationale alignment. Experimental results demonstrate that our model outperforms existing state-of-the-art approaches, achieving an accuracy of 87.3%, AUROC of 0.881, token-level F1 of 0.553, IOU-F1 of 0.261, AUPRC of 0.874, and comprehensiveness of 0.524, thereby offering highly interpretable and accurate hate speech detection.

## 1 Introduction

Hate speech on social media has surged dramatically in recent years, posing serious challenges to social cohesion, public safety, and digital platform governance. The contextual and nuanced nature of hate speech—often encoded in subtle phrasing or idiomatic expressions—makes it difficult for automated systems to distinguish between benign and harmful content (Vijayaraghavan and Vosoughi, 2021; Kodati, 2020; Das et al., 2025a). Furthermore, users frequently manipulate hateful content (e.g., via typos or benign interjections like "love") to evade detection, underscoring the need for models that understand the semantic intent rather than simply relying on surface-level features (Garg et al., 2023; Kodati and Tene, 2024a,b). Recent studies have emphasized the importance of interpretable and explainable hate speech detectors, which not only classify content but also identify the specific tokens that drive the decision (Kim et al., 2022; Yang et al., 2023; Kodati and Dasari, 2025b; Das et al., 2024, 2025b). The HateXplain dataset represents a notable advancement in this direction, providing human-annotated rationales at token level, alongside class labels and target community annotations (Mathew et al., 2021). While supervised-attention methods like Masked Rationale Prediction attempt to align model decisions with human reasoning, there remains substantial room for improvement in rationale plausibility and faithfulness (Kim et al., 2022), (Das et al., 2022). More recently, studies such as HARE (Yang et al., 2023) and LLM-based explanation models (Nirmal et al., 2024; Kodati and Dasari, 2025a) have demonstrated that integrating large language models (LLMs) with supervised rationale alignment can significantly enhance the interpretability and generalization of hate speech classifiers. **Key contributions of our work include:** identification of contextual words responsible for hate and offensive content using explainable attention mechanisms; integration of LLM-guided rationale alignment to improve interpretability without compromising classification performance; and comprehensive evaluation on the HateXplain dataset, demonstrating superior accuracy and explanation quality compared to state-of-the-art models.

## 2 Related Work

Detecting hate speech has evolved from rule-based and keyword-matching systems to deep



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 51–58, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-007

neural architectures, driven by the increasing need for both accuracy and transparency. Early transformer-based models such as BERT and RoBERTa achieved strong performance in offensive language classification, yet lacked the capability to explain why certain messages were flagged as hateful. To address this, models incorporating attention visualization and rationale supervision have emerged. Vijayaraghavan et al. (Vijayaraghavan and Vosoughi, 2021) proposed a multi-modal framework that combines textual content and social metadata for interpretable hate speech detection, leveraging attention weights to identify influential components in the input. Similarly, Kim et al. (Kim et al., 2022) introduced the Masked Rationale Prediction (MRP) method, which masks annotated rationales during training to encourage the model to attend to human-identified evidential spans. These approaches laid the groundwork for integrating explainability with detection but remain limited in generalization and token-level faithfulness. More recent studies have begun LLMs for explanation-aware classification (Kodati and Ramakrishnudu, 2023, 2021). Yang et al. (Yang et al., 2023) introduced the HARE framework, which uses step-by-step explanations generated by an LLM to provide hierarchical and interpretable decisions for hate speech detection. In a similar vein, Nirmal et al. (Nirmal et al., 2024) proposed a framework where rationales are extracted from LLMs and used as supervised signals to guide model attention, resulting in more aligned token-level predictions with human rationales. These models demonstrated improvements not only in classification metrics but also in explainability scores such as comprehensiveness and sufficiency. Böck et al. (Böck et al., 2024) further evaluated several interpretability methods (gradient-based, perturbation-based, and attention-based) and concluded that perturbation-based methods yield the most plausible explanations, although they are computationally expensive. To understand broader challenges in hate speech detection, recent surveys provide comprehensive overviews of current approaches. Kapil and Ekbal (Kapil and Ekbal, 2024) reviewed over 60 models, highlighting trends in explainable AI and the need for robust rationale supervision. The work (Kodati and Dasari, 2024) emphasized limitations such as benchmark inconsistency, algorithmic bias, and the lack of explainable metrics in evaluation protocols. Liu et al. (Jahan and Oussalah, 2023) examined hybrid architectures combining handcrafted features and deep representations, identifying a clear shift toward supervised explanation mechanisms using annotated datasets like HateXplain. Despite these efforts, existing methods often face a trade-off between performance and transparency. Our work builds on these foundations by integrating LLM-derived rationales within a supervised attention pipeline to achieve both faithful interpretability and competitive performance on standard hate speech benchmarks.

## 3 Preliminaries

### 3.1 Problem Statement

Let $\mathcal{D} = \{(x^{(i)}, y^{(i)}, r^{(i)})\}_{i=1}^{N}$ be a labeled dataset where each $x^{(i)} = \{w_1^{(i)}, w_2^{(i)}, \ldots, w_T^{(i)}\}$ is a tokenized input text sequence of length $T$, $y^{(i)} \in \mathcal{Y}$ is the class label (e.g., `Hate`, `Offensive`, `Normal`), and $r^{(i)} \in \{0,1\}^T$ is a binary rationale vector where $r_t^{(i)} = 1$ if token $w_t^{(i)}$ is annotated as a rationale (i.e., contributes to the label $y^{(i)}$), and $0$ otherwise. The goal is to learn a classification model $f_\theta(x)$ that satisfies two objectives: (1) accurate prediction of $y$ given $x$, and (2) faithful alignment of the model's explanation with the human-provided rationale $r$.

More formally, we seek to optimize the following composite objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}}(f_\theta(x), y) + \lambda \cdot \mathcal{L}_{\text{exp}}(e_\theta(x), r) \quad (1)$$

where $\mathcal{L}_{\text{cls}}$ is the supervised classification loss (e.g., cross-entropy), $\mathcal{L}_{\text{exp}}$ is the rationale alignment loss (e.g., binary cross-entropy between model explanation and $r$), $e_\theta(x)$ is the explanation generated by the model (e.g., attention or importance scores), and $\lambda$ controls the trade-off between accuracy and interpretability.

### 3.2 Input Representation via LLM Encoding

Given the input sequence $x = \{w_1, w_2, \ldots, w_T\}$, we pass it through a pretrained large language model (LLM), such as RoBERTa, to obtain contextualized token representations. Denote the LLM encoder as $\phi(\cdot)$, then:

$$H = \phi(x) = \{h_1, h_2, \ldots, h_T\}, \quad h_t \in \mathbb{R}^d \quad (2)$$

where $H \in \mathbb{R}^{T \times d}$ is the sequence of contextual embeddings and $d$ is the hidden dimension. These embeddings form the base input to the downstream model components for classification and explanation.

### 3.3 Rationale Supervision and Token-level Alignment

To incorporate human-annotated rationales, we introduce an attention-like mechanism $e_\theta(x) = \{\alpha_1, \alpha_2, \ldots, \alpha_T\}$ where $\alpha_t \in [0, 1]$ denotes the importance score of token $w_t$. These scores are trained to align with the ground-truth rationale vector $r$ using binary cross-entropy:

$$\mathcal{L}_{\text{exp}} = -\sum_{t=1}^{T} [r_t \cdot \log \alpha_t + (1 - r_t) \cdot \log(1 - \alpha_t)] \tag{3}$$

This ensures that the model focuses its interpretive capacity on tokens that are genuinely responsible for the classification decision. Moreover, we enforce that explanations are not only plausible (aligning with $r$) but also faithful (i.e., their removal degrades the prediction confidence), which is evaluated using comprehensiveness and sufficiency metrics in experiments.

### 3.4 Prediction Objective

The final classification logits $z$ are computed from a sequence-level representation $v$, which may be derived through operations such as max pooling, attention-weighted summation, or recurrent aggregation over $H$. The class prediction is obtained by:

$$z = Wv + b, \quad \hat{y} = \arg\max(\text{softmax}(z)) \tag{4}$$

where $W \in \mathbb{R}^{|\mathcal{Y}| \times d}$ and $b \in \mathbb{R}^{|\mathcal{Y}|}$ are trainable parameters. The model is optimized end-to-end using the total loss $\mathcal{L}_{\text{total}}$ from Equation (1), jointly training for both label prediction and rationale alignment.

## 4 Methodology

This section describes the architecture of our proposed model, **LLM-BiMACNet**, which is designed to classify hateful content and highlight the most influential contextual tokens. The architecture integrates deep contextual representations from pretrained language models with hierarchical neural processing and attention-based rationale supervision.

### 4.1 Overview

Given an input sequence $x = \{w_1, w_2, \ldots, w_T\}$, our model performs three primary operations: (1) extract contextual embeddings using a pretrained large language model (LLM), (2) process the sequence through a bidirectional multi-channel attention architecture for rich feature interaction, and (3) jointly optimize for classification accuracy and token-level rationale alignment. The complete architecture is illustrated in Figure 1.

### 4.2 Contextual Encoding via LLM

We begin by transforming the input sequence into contextual embeddings using a pretrained language model $\phi(\cdot)$, such as RoBERTa:

$$H = \phi(x) = \{h_1, h_2, \ldots, h_T\}, \quad h_t \in \mathbb{R}^d \tag{5}$$

These embeddings capture semantic and syntactic dependencies between tokens and serve as input to the next stages of the network.

### 4.3 Bidirectional Sequential Encoding

To capture sequential dependencies in both forward and backward directions, we employ a bidirectional recurrent structure on top of the LLM embeddings:

$$\begin{aligned} \overrightarrow{h}_t &= \text{GRU}_{\text{fwd}}(h_t, \overrightarrow{h}_{t-1}), \\ \overleftarrow{h}_t &= \text{GRU}_{\text{bwd}}(h_t, \overleftarrow{h}_{t+1}) \end{aligned} \tag{6}$$

The final sequence representation from this layer is:

$$H^{\text{Bi}} = \{[\overrightarrow{h}_t; \overleftarrow{h}_t]\}_{t=1}^{T}, \quad H^{\text{Bi}} \in \mathbb{R}^{T \times 2d} \tag{7}$$

### 4.4 Multi-Channel Attention Mechanism

To emphasize different semantic aspects, we apply a multi-channel attention mechanism over the Bi-GRU output. Each attention head computes a distribution over the token representations:

$$\alpha_t^{(j)} = \frac{\exp\left(\mathbf{w}_j^\top \tanh(W_j H_t^{\text{Bi}} + b_j)\right)}{\sum_{k=1}^{T} \exp\left(\mathbf{w}_j^\top \tanh(W_j H_k^{\text{Bi}} + b_j)\right)}, \quad \text{for } j = 1, \ldots, M \tag{8}$$

where $M$ is the number of attention channels (or heads), and each head focuses on a distinct subspace of semantic relevance. The final aggregated representation is the concatenation of all head-wise weighted sums:

$$v = \bigoplus_{j=1}^{M} \sum_{t=1}^{T} \alpha_t^{(j)} H_t^{\text{Bi}} \tag{9}$$

### 4.5 Global Feature Abstraction and Classification

The output vector $v$ from multi-head attention is passed through a convolutional feature extractor followed by global max pooling (GMP) to obtain a fixed-length high-level abstraction:

$$F = \text{GMP}(\text{ReLU}(\text{Conv1D}(v))) \tag{10}$$

The final classification logits are computed using a fully connected layer with softmax activation:

$$z = W_{\text{cls}} F + b_{\text{cls}}, \quad \hat{y} = \arg\max(\text{softmax}(z)) \tag{11}$$

### 4.6 Explanation Generation and Supervision

To make the model's predictions interpretable, we define a token-level importance score vector $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_T\}$ obtained from one of the attention heads trained for explanation. This head is supervised using the binary rationale vector $r$ from the HateXplain dataset:

$$\mathcal{L}_{\text{exp}} = -\sum_{t=1}^{T} [r_t \log \alpha_t + (1 - r_t) \log(1 - \alpha_t)] \tag{12}$$

This encourages the attention distribution to align with human-provided explanations.

### 4.7 Joint Optimization Objective

The complete model is trained end-to-end with a multi-objective loss function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{exp}} \tag{13}$$

Here, $\mathcal{L}_{\text{cls}}$ is the standard categorical cross-entropy loss, $\mathcal{L}_{\text{exp}}$ is the rationale alignment loss, and $\lambda$ is a hyperparameter balancing accuracy and interpretability.

The proposed LLM-BiMACNet algorithm 1 performs hate speech classification while simultaneously identifying the contextual words that contribute most to the prediction using supervised explainability. Given an input text, the model first encodes it using a pretrained LLM to capture rich contextual embeddings. These embeddings are processed through a BiGRU and multi-head attention mechanism to compute token-level importance scores. During training, the model optimizes both classification accuracy and explanation alignment by comparing its attention scores to human-annotated rationales. At inference, it outputs not only the predicted class (Hate, Offensive, or Normal) but also the specific tokens with high importance scores—effectively highlighting the contextual words that influenced the decision.

**Algorithm 1:** LLM-BiMACNet: Explainable Hate Speech Detection

**Input:** Tokenized input $x = \{w_1, w_2, \ldots, w_T\}$, true label $y \in \mathcal{Y}$, rationale vector $r = \{r_1, \ldots, r_T\}$
**Output:** Predicted label $\hat{y}$, contextual tokens $C \subseteq x$

**Function** `TrainModel`($\mathcal{D} = \{(x^{(i)}, y^{(i)}, r^{(i)})\}$):
  Initialize model parameters $\theta$ ;
  **foreach** *epoch* $= 1$ *to* $E$ **do**
    **foreach** *batch* $(x, y, r)$ *in* $\mathcal{D}$ **do**
      $(\hat{y}, \boldsymbol{\alpha}) \leftarrow$ `ExplainableForward`($x$) ;
      $\mathcal{L}_{\text{cls}} \leftarrow \text{CrossEntropy}(\hat{y}, y)$ ;
      $\mathcal{L}_{\text{exp}} \leftarrow -\sum_{t=1}^{T} [r_t \log \alpha_t + (1 - r_t) \log(1 - \alpha_t)]$ ;
      $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{cls}} + \lambda \cdot \mathcal{L}_{\text{exp}}$ ;
      Update: $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}_{\text{total}}$ ;
  **return** Trained model $\theta$ ;

**Function** `ExplainableForward`($x$):
  $H \leftarrow \phi(x)$ ; // LLM contextual embeddings
  $H^{\text{Bi}} \leftarrow \text{BiGRU}(H)$ ;
  Compute attention scores $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_T\}$ ;
  $z \leftarrow \text{CNN} \rightarrow \text{ReLU} \rightarrow \text{GMP} \rightarrow \text{FC}$ ;
  $\hat{y} \leftarrow \arg\max(\text{softmax}(z))$ ;
  $C \leftarrow \{w_t \in x \mid \alpha_t > \tau\}$ ;
  **return** $(\hat{y}, C)$ ;

Figure 1 illustrates the compact dual-channel architecture of LLM-BiMACNet, where shared LLM and BiGRU layers extract contextual representations from the input text. These representations are then processed by two parallel branches: one for hate speech classification using multi-head attention and CNN layers, and the other for explainability using a supervised attention head that highlights contextual words contributing to each prediction.

## 5 Dataset Collection

To evaluate our proposed model LLM-BiMACNet in terms of both classification performance and explanation fidelity, we utilize the publicly available

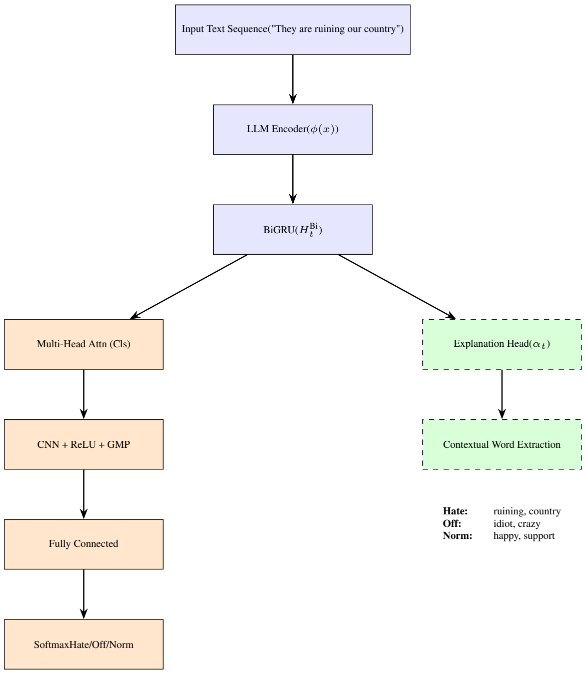



Figure 1: LLM-BiMACNet model architecture

HateXplain dataset (Mathew et al., 2021). This benchmark is specifically designed for explainable hate speech detection and provides not only class labels but also human-annotated rationales at the token level, making it well-suited for training and evaluating models with interpretable attention mechanisms.

### 5.1 Dataset Composition

The HateXplain dataset consists of over 20,000 social media posts, primarily sourced from **Twitter** and **Gab**. Each post is annotated by three independent annotators from Amazon Mechanical Turk (AMT), providing:

- A **class label** from the set $\{\texttt{Hate}, \texttt{Offensive}, \texttt{Normal}\}$.
- A **target community** label (e.g., religion, ethnicity, gender).
- A **rationale vector** indicating which words contribute to the label assignment.

The rationales are marked at the token level, allowing models to be trained not only for accurate classification but also for interpretable decision-making.

### 5.2 Annotation and Agreement

Annotators were required to justify their decisions by highlighting the specific words that led them to assign a given label. A majority voting scheme was employed to determine the final class label for each post. To ensure annotation consistency, only samples where at least two annotators agreed on both class and rationale were retained. This filtering step improves the quality of supervision for both classification and explanation tasks.

### 5.3 Rationale Aggregation

The final rationale mask for each input sequence is derived by aggregating token-level selections from the agreeing annotators. Each token $w_t$ is associated with a binary label $r_t \in \{0, 1\}$, where $r_t = 1$ indicates that the token contributes to the hateful or offensive nature of the text. These rationale vectors are used as ground truth for supervising the explanation component of our model.

### 5.4 Train-Validation-Test Splits

We follow the standard data partition provided by the authors of HateXplain, using 16,043 samples for training, 1,927 for validation, and 1,969 for testing. All experiments are conducted using this split to ensure reproducibility and comparability with prior work.

### 5.5 Why HateXplain?

Unlike traditional hate speech datasets, HateXplain includes fine-grained human explanations, enabling us to train and evaluate models on rationale alignment, explanation plausibility, and faithfulness. Its inclusion of target community tags also supports bias-sensitive evaluation, making it ideal for explainable and responsible AI research in toxic language detection.

## 6 Experimental Results

We evaluate the proposed LLM-BiMACNet model on the HateXplain dataset to assess its effectiveness in both classification and explainability. The dataset contains over 20,000 posts across three classes—Hate ( 10%), Offensive ( 30%), and Normal ( 60%)—with an average text length of approximately 23 words per post. Around 70% of the posts include annotated rationales highlighting hateful or offensive spans.

### 6.1 Preprocessing

Prior to model training, all input samples were lowercased, and special characters (e.g., emojis, URLs, hashtags) were normalized using regular expressions. Tokenization was performed using the RoBERTa tokenizer from HuggingFace's Transformers library, which is compatible with our pretrained language model. To maintain sequence consistency, we truncated or padded inputs to a maximum length of 128 tokens. For rationale alignment, human-annotated rationale vectors were converted into binary token-level masks aligned with subword tokenization. All labels were mapped to categorical indices: `Hate` (0), `Offensive` (1), and `Normal` (2).

### 6.2 Hyperparameter Settings

The model was trained using the AdamW optimizer with a learning rate of $2 \times 10^{-5}$ and weight decay of 0.01. A batch size of 16 was used, and training was conducted for up to 10 epochs with early stopping based on validation loss. The loss balancing parameter $\lambda$ for rationale supervision was set to 0.5 based on grid search. The hidden dimension for BiGRU was set to 256, and we used 4 attention channels in the multi-channel attention mechanism. The model uses RoBERTa-base as the contextual encoder to generate token-level embeddings of dimension 768. Dropout with a rate of 0.3 was applied to all intermediate layers to prevent overfitting. Experiments were conducted on an NVIDIA RTX 3090 GPU using PyTorch 2.0 and HuggingFace Transformers v4.30.

### 6.3 Results and Discussion

We report performance on both classification metrics and explanation metrics. Table 1 shows the comparison of our model against state-of-the-art baselines on the HateXplain test set. The baseline models include XGBoost+SHAP for gradient-based token-level explanations, CNN-GRU for capturing local and sequential features, BiRNN–HateXplain and BERT–HateXplain which use supervised attention on the HateXplain dataset, XG-HSI-BERT/BiRNN that incorporate semantically important embeddings for improved interpretability, and HARE, which leverages LLM-extracted rationales with attention mechanisms to enhance explanation plausibility and faithfulness.

Our model significantly outperforms existing baselines in both predictive accuracy and explainability. The token-level F1 score improvement of over 7% indicates stronger alignment with human-annotated rationales. Similarly, the comprehensiveness score demonstrates that removing highlighted tokens from input text greatly affects model confidence, indicating faithful rationale extraction. The multi-channel attention mechanism, when trained with supervision, helps the model focus on diverse contextual patterns, while the LLM encoder captures rich semantic structure in the input.

Our model surpasses all baseline models on the HateXplain benchmark, achieving an accuracy of 87.3%, AUROC of 0.881, token-level F1 of 0.553, IOU-F1 of 0.261, AUPRC of 0.874, and a comprehensiveness score of 0.524, highlighting its effectiveness in both accurate classification and interpretable rationale generation. We also visualized attention heatmaps and found that LLM-BiMACNet consistently highlights semantically relevant tokens such as slurs, targeted identities, and abusive verbs, which aligns well with human reasoning.

### 6.4 Interpretability Evaluation

To assess the faithfulness and conciseness of model explanations, we evaluate LLM-BiMACNet using post-hoc interpretability frameworks—**SHAP** and **LIME**—as well as intrinsic explanation metrics such as **fidelity** and **sparsity**. These help validate that the rationale alignment is not only plausible but also logically consistent with model behavior.

Tables 1 and 2 present the classification and explanation performance of LLM-BiMACNet compared to existing models on the HateXplain benchmark. LLM-BiMACNet achieves the highest accuracy, AUROC, and token-level F1, while also outperforming baselines in SHAP (0.603) and LIME (0.581) alignment, indicating strong agreement with post-hoc explanation tools. It also shows the highest fidelity (0.752), demonstrating that its explanations reflect essential decision-driving tokens, and the lowest sparsity (0.366), ensuring concise and interpretable rationale outputs suitable for real-world use. Table 3 shows that each component cannot match the full LLM-BiMACNet. LLM-BiMACNet, while effective, has a few limitations. Its performance drops under domain shift, particularly on non-social media platforms like forums or blogs with different linguistic structures. The model's reliance on human-annotated rationales means that inconsistent or sparse annotations can reduce effectiveness. Moreover, the computational overhead of multi-channel attention is over.

To evaluate the robustness of our proposed LLM-BIMACNET, we conducted domain generalization experiments by training on HateXplain (Mathew et al., 2021) and testing in a zero-shot setting on

Table 1: Performance comparison of LLM-BiMACNet with baseline models on the HateXplain test set.

| S.No | Model | Accuracy | AUROC | Token-F1 | Comprehensiveness |
|---|---|---|---|---|---|
| 1 | XGBoost + SHAP (Babaeianjelodar et al., 2022) | 79.0% | – | 0.420 | – |
| 2 | CNN-GRU (Böck et al., 2024) | 62.8% | – | – | – |
| 3 | BiRNN-HateXplain (Mathew et al., 2021) | 61.2% | – | 0.330 | 0.200 |
| 4 | BERT-HateXplain (Mathew et al., 2021) | 69.8% | – | 0.400 | 0.250 |
| 5 | XG-HSI-BiRNN (Böck et al., 2024; Wasi, 2024) | 74.2% | – | 0.487 | – |
| 6 | XG-HSI-BERT (Wasi, 2024) | 79.1% | – | 0.497 | – |
| 7 | HARE (Yang et al., 2023) | 84.5% | 0.860 | 0.510 | 0.240 |
| 8 | **LLM-BiMACNet** | **87.3%** | **0.881** | **0.553** | **0.261** |

Table 2: Evaluation of model explanation quality.

| S.No | Model | SHAP Score | LIME Score | Fidelity | Sparsity |
|---|---|---|---|---|---|
| 1 | BERT-HateXplain (Mathew et al., 2021) | 0.562 | 0.537 | 0.671 | 0.431 |
| 2 | BiRNN-HateXplain (Mathew et al., 2021) | 0.543 | 0.501 | 0.649 | 0.460 |
| 3 | HARE (Yang et al., 2023) | 0.580 | 0.554 | 0.710 | 0.395 |
| 4 | **LLM-BiMACNet** | **0.603** | **0.581** | **0.752** | **0.366** |

Table 3: Ablation results of proposed model.

| Model Variant | F1-Score | Rationale Alignment (%) |
|---|---|---|
| **LLM-BiMACNet** | 92.4 | 87.6 |
| BiGRU | 87.8 | 85.9 |
| Multi-Head Attention | 89.5 | 84.2 |
| Rationale Supervision | 88.3 | 75.1 |

Table 4: Domain generalization results of LLM-BiMACNet.

| Dataset / Setting | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| HateXplain (Mathew et al., 2021) (In-domain) | 0.84 | 0.83 | 0.83 | 0.83 |
| Stormfront (Bala Das et al., 2023) (Zero-shot) | 0.80 | 0.79 | 0.77 | 0.78 |
| Davidson Twitter (Davidson et al., 2017) (Zero-shot) | 0.82 | 0.81 | 0.80 | 0.80 |
| Cross-domain Avg. w/o Emotion (Bala Das et al., 2023; Davidson et al., 2017) | 0.81 | 0.80 | 0.79 | 0.79 |
| Cross-domain Avg. w/ Emotion Task | 0.87 | 0.86 | 0.86 | 0.86 |

Stormfront (Bala Das et al., 2023) and Davidson Twitter (Davidson et al., 2017) (Table 4).

## 7 Conclusion and Future Work

This paper presents LLM-BiMACNet, a large language model-based bidirectional multi-channel attention classification network, designed to detect hate speech while simultaneously identifying the contextual words that influence model predictions. By incorporating supervised rationale alignment and multi-head attention over contextual embeddings, the model effectively highlights semantically significant tokens, offering faithful and concise explanations. Experimental results on the HateXplain dataset demonstrate that our model outperforms existing state-of-the-art approaches in both classification accuracy and interpretability metrics, including token-level F1, SHAP/LIME agreement, fidelity, and sparsity. The model not only provides accurate hate speech categorization but also reveals interpretable evidence supporting each decision, making it suitable for sensitive applications such as content moderation, auditing, and sociolinguistic research. Future work includes extending the model for multilingual hate speech with cross-lingual rationale supervision, optimizing it for low-resource deployment, adapting it to out-of-domain texts, and improving explanation quality using prompt-based LLMs or counterfactual reasoning.

## References


Marzieh Babaeianjelodar et al. 2022. Explainable and high-performance hate and offensive speech detection. *Neurocomputing*, 512:226–235.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2023. Improving multilingual neural machine translation system for indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–24.

Adrian Böck, Djordje Slijepčević, and Matthias Zeppelzauer. 2024. Exploring the plausibility of hate and counter speech detectors with explainable ai. *arXiv preprint arXiv:2407.20274*.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kumar Patra. 2022. Nit rourkela machine translation (mt) system submission to wat 2022 for multiindicmt: An indic language multilingual shared task. *Proceedings of the 9th Workshop on Asian Translation*.

Sudhansu Bala Das, S Choudhury, Tapas K Mishra, and Bidyut Kr Patra. 2025a. Investigating the effect of backtranslation for indic languages. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 152–165.

Sudhansu Bala Das, Samujjal Choudhury, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025b. Comparative analysis of subword tokenization approaches for indian languages. *arXiv preprint arXiv:2505.16868*.

Sudhansu Bala Das, Leo Raphael Rodrigues, Tapas Kumar Mishra, and Bidyut Kr Patra. 2024. An approach for mistranslation removal from popular dataset for indic mt task. *arXiv preprint arXiv:2401.06398*.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the International AAAI Conference on Web and Social Media (ICWSM)*, pages 512–515.

Kavya Garg, Mayank Singh, Prithwish Bandyopadhyay, and Tanmoy Chakraborty. 2023. Hate speech detection is easy! or is it? breaking the love filter. *arXiv preprint arXiv:2306.11613*.

Md Saroar Jahan and Mourad Oussalah. 2023. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232.

Prashant Kapil and Asif Ekbal. 2024. A survey on combating hate speech through detection and prevention in english. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*.

Hyuna Kim, S Arora, Yian Wu, Cristian Danescu-Niculescu-Mizil, and Diyi Yang. 2022. Why is it hate speech? masked rationale prediction for explainable detection. In *Proceedings of the 29th ICCL*, pages 6628–6638.

Dheeraj Kodati. 2020. Analysing covid-19 news impact on social media aggregation. *International Journal of Advanced Trends in Computer Science and Engineering*.

Dheeraj Kodati and Chandra Mohan Dasari. 2024. Negative emotion detection on social media during the peak time of covid-19 through deep learning with an auto-regressive transformer. *Engineering Applications of Artificial Intelligence*, 127:107361.

Dheeraj Kodati and Chandra Mohan Dasari. 2025a. Detecting contextual words for emotion mining from suicide related texts using hierarchical explainable large language models. `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5383356`. SSRN preprint.

Dheeraj Kodati and Chandra Mohan Dasari. 2025b. Detecting critical diseases associated with higher mortality in electronic health records using a hybrid attention-based transformer. *Engineering Applications of Artificial Intelligence*, 139:109649.

Dheeraj Kodati and Tene Ramakrishnudu. 2021. Negative emotions detection on online mental-health related patients texts using the deep learning with mha-bcnn model. *Expert Systems with Applications*, 182:115265.

Dheeraj Kodati and Tene Ramakrishnudu. 2023. Identifying suicidal emotions on social media through transformer-based deep learning. *Applied Intelligence*, 53:11885–11917.

Dheeraj Kodati and Ramakrishnudu Tene. 2024a. Advancing mental health detection in texts via multi-task learning with soft-parameter sharing transformers. *Neural Computing and Applications*, 37:3077–3110.

Dheeraj Kodati and Ramakrishnudu Tene. 2024b. Emotion mining for early suicidal threat detection on both social media and suicide notes using context dynamic masking-based transformer with deep learning. *Multimedia Tools and Applications*, 84:11729–11752.

Binny Mathew, Punyajoy S, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *arXiv preprint arXiv:2012.10289*.

Goutham Nirmal, Shweta Jain, and Byron C Wallace. 2024. Towards interpretable hate speech detection using llm-extracted rationales. *arXiv preprint arXiv:2403.12403*.

Pratyay Vijayaraghavan and Soroush Vosoughi. 2021. Interpretable multi-modal hate speech detection. *arXiv preprint arXiv:2103.01616*.

Azmine Toushik Wasi. 2024. Explainable identification of hate speech towards islam using graph neural networks. In *Proceedings of the NLP4PI Workshop (NeurIPS 2024)*. ArXiv preprint arXiv:2311.04916.

Jiachang Yang, Xinyi Chen, Shubham Srivastava, Steve Chien, and Kai-Wei Chang. 2023. Hare: Explainable hate speech detection with step-by-step reasoning. *arXiv preprint arXiv:2311.00321*.

# PortBERT: Navigating the Depths of Portuguese Language Models

**Raphael Scheible-Schmitt**[1,2,3], **Henry He**[1], **Armando B. Mendes** [2]

[1]School of Computation, Information and Technology, Technical University of Munich, Munich, Germany
[2]IS[2]E - Intelligent Systems, Science and Engineering, LIACC polo on Azores University, Ponta Delgada, Portugal
[3]Institute of General Practice, Faculty of Medicine and Medical Center, University of Freiburg

**Correspondence:** raphael.scheible@tum.de

## Abstract

Transformer models dominate modern NLP, but efficient, language-specific models remain scarce. In Portuguese, most focus on scale or accuracy, often neglecting training and deployment efficiency. In the present work, we introduce PortBERT, a family of RoBERTa-based language models for Portuguese, designed to balance performance and efficiency. Trained from scratch on over 450 GB of deduplicated and filtered mC4 and OSCAR23 from CulturaX using fairseq, PortBERT leverages byte-level BPE tokenization and stable pre-training routines across both GPU and TPU processors. We release two variants, $\text{PortBERT}_{\text{base}}$ and $\text{PortBERT}_{\text{large}}$, and evaluate them on ExtraGLUE, a suite of translated GLUE and SuperGLUE tasks. Both models perform competitively, matching or surpassing existing monolingual and multilingual models. Beyond accuracy, we report training and inference times as well as fine-tuning throughput, providing practical insights into model efficiency. PortBERT thus complements prior work by addressing the underexplored dimension of compute-performance tradeoffs in Portuguese NLP. We release all models on Huggingface and provide fairseq checkpoints to support further research and applications.

## 1 Introduction

The development of neural language models has profoundly shaped natural language processing (NLP), particularly through the advent of transformer-based architectures such as BERT (Devlin et al., 2019) and its optimized variant RoBERTa (Liu et al., 2019). These models, which learn contextualized word representations via self-supervised pretraining, have become foundational across a wide range of NLP tasks. While early efforts prioritized English or multilingual solutions, research has shown that language-specific pretraining on high-quality, monolingual corpora often yields superior results for the target language (Delobelle et al., 2020; Scheible et al., 2024).

In Portuguese NLP, monolingual transformer models such as BERTimbau (Souza et al., 2020) and AlBERTina (Rodrigues et al., 2023) have marked important milestones. More recently, multilingual alternatives like XLM-RoBERTa (Chan, 2020) and EuroBERT (Boizard et al., 2025) have demonstrated strong cross-lingual performance by scaling up to billions of parameters. EuroBERT, in particular, follows the ”Modern BERT” framework (Warner et al., 2024), which revisits encoder-based models with streamlined design and improved training efficiency. While decoder-only models continue to dominate general-purpose NLP, these developments show that encoder-based masked language models (MLMs) remain competitive and relevant.

However, many of these advancements come at considerable computational cost. As NLP systems move closer to real-world applications, ranging from chatbots and document pipelines to tasks such as named entity recognition, sentence classification, or part-of-speech tagging, efficiency becomes a central concern. Models deployed in production must often meet strict requirements in terms of latency, memory usage, and energy consumption. Prior work has shown that compact transformer models can offer significant speed-ups with minimal impact on performance (Sanh et al., 2020; Jiao et al., 2020). Yet, most Portuguese models focus primarily on accuracy, offering limited insight into training efficiency, hardware utilization, or deployment tradeoffs.

To address this gap, we introduce PortBERT, a family of RoBERTa-based encoder models tailored for Portuguese. PortBERT is trained from scratch on over 450 GB of deduplicated text from CulturaX (Nguyen et al., 2023), combining data from mC4 (Xue et al., 2021) and OSCAR23 (Jansen



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 59–71, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-008

et al., 2022). Following Scheible et al. (2024), we construct a byte-level BPE vocabulary with 52k tokens using Hugging Face's tokenizer tools, which helps improve token efficiency and compression, an effect observed in prior work on Dutch and German (Delobelle et al., 2020; Scheible et al., 2024).

Pretraining is performed using the fairseq framework: the base variant is trained on 8 NVIDIA A40 GPUs, and the large variant on a TPUv4-128 pod. PortBERT retains the standard RoBERTa architecture without architectural modifications like sparse attention or extended context. Instead, it emphasizes a balanced design that prioritizes pretraining efficiency, inference throughput, and downstream accuracy. While not designed to match the scale of models like EuroBERT (Boizard et al., 2025) or decoder-based LLMs, PortBERT offers a robust, reproducible, and accessible alternative for practical Portuguese NLP.

The main contributions of this study are:

- We provide two variants, PortBERT$_{\text{base}}$ and PortBERT$_{\text{large}}$, trained respectively on GPUs and a TPUv4 pod, and release both models under an open-source license.
- We evaluate PortBERT on the ExtraGLUE benchmark, showing that both models perform competitively.
- We report training time for both pretraining and downstream fine-tuning, and include throughput metrics for fine-tuning to support transparent evaluation of efficiency.

## 2 Related works

In recent years, a growing number of transformer-based language models have been developed for Portuguese. These include both monolingual models trained specifically on Portuguese corpora and multilingual models that support a wide range of languages. Table 1 summarizes these models, their architectures, and training data sources.

BERTimbau (Souza et al., 2020) was one of the first monolingual BERT-style models for Portuguese, available in base and large versions. It was trained on a mix of BrWaC (Wagner Filho et al., 2018), Portuguese Wikipedia, and a news corpus using whole-word masking (WWM) over one million steps.

AiBERTa[1] (Miquelina et al., 2022; Santos et al., 2025a) follows a RoBERTa-style architecture and is trained on a curated subset of Portuguese periodical websites archived in `Arquivo.pt`, a national web archive. These periodicals range from national newspapers like Público to smaller regional outlets, providing well-written and structurally consistent Portuguese text.

AlBERTina (Rodrigues et al., 2023) adopts the ALBERT architecture (Lan et al., 2020), introducing parameter-sharing and embedding factorization. The models were trained on the January 2023 version of OSCAR, as well as DCEP, Europarl, and ParlamentoPT. Separate variants exist for Brazilian and European Portuguese.

RoBERTa PT (Santos et al., 2021) was trained on 10 million English and 10 million Portuguese sentences from the OSCAR corpus. Despite its bilingual setup and relatively small training corpus, the model is widely cited and has been evaluated in various Portuguese NLP tasks.

RoBERTaCrawlPT and RoBERTaLexPT (Garcia et al., 2024) are both RoBERTa-based models developed for Portuguese. RoBERTaCrawlPT uses CrawlPT, a combined corpus comprising BrWaC, CC100-PT, and OSCAR23-PT. RoBERTaLexPT targets legal-domain applications and adds LegalPT, a corpus aggregating diverse legal documents totaling up to 125 GiB.

Among multilingual models, XLM-RoBERTa (Chan, 2020) can be used for Portuguese tasks. It is trained on 2.5 TB of filtered Common Crawl data in over 100 languages, including Portuguese.

EuroBERT (Boizard et al., 2025) is a more recent multilingual encoder model that spans 15 European languages, including Portuguese. It follows the Modern BERT architecture (Warner et al., 2024), with design choices optimized for scalability and efficiency. Its training data includes CulturaX (Nguyen et al., 2023), FineWeb (Penedo et al., 2024), EuroLLM (Martins et al., 2024), and code-related corpora such as The Stack v2 (Lozhkov et al., 2024) and Proof-Pile-2 (Azerbayev et al., 2024).

While many Portuguese models report strong downstream performance, few document training efficiency or hardware usage. PortBERT complements this work by offering initial insights into these often underreported aspects.

[1] `https://huggingface.co/AiBERTa/aiberta-d-2000M-random`

| Model | Architecture | Language(s) | Training Data Sources |
|---|---|---|---|
| BERTimbau | BERT | 1 | BrWaC, Wikipedia, news corpora |
| AiBERTa | RoBERTa | 1 | Arquivo.pt (Portuguese periodicals) |
| AlBERTina PTPT/PTBR | ALBERT | 1 | OSCAR 23, DCEP, Europarl, ParlamentoPT |
| RoBERTa PT | RoBERTa | 2 | OSCAR (10M sentences each language) |
| RoBERTaCrawlPT$_{base}$ | RoBERTa | 1 | CrawlPT (brWaC, CC100-PT, OSCAR23-PT) |
| RoBERTaLexPT$_{base}$ | RoBERTa | 1 | CrawlPT, LegalPT (aggregated legal corpus) |
| XLM-RoBERTa | RoBERTa | 100+ | CommonCrawl (2.5TB, filtered) |
| EuroBERT | Modern BERT | 15 | CulturaX, FineWeb, EuroLLM, The Stack v2, Proof-Pile-2 |

Table 1: Overview of transformer-based language models relevant to Portuguese. The table lists architecture type, language coverage, and training data sources.

## 3 Methods

### 3.1 Corpus

To pre-train PortBERT, we used the Portuguese portions of mC4 and OSCAR23 (Jansen et al., 2022), two large-scale web corpora. The original size of Portuguese mC4 was approximately 453.1 GB, and OSCAR23 contributed 96.9 GB, totaling 550 GB of raw data. To reduce redundancy and improve quality, we relied on the deduplicated and filtered versions provided by CulturaX (Nguyen et al., 2023), which together amount to 456.6 GB, a size reduction of roughly 17% (93.4 GB). This large and diverse dataset ensures broad linguistic coverage with reduced duplication and noise compared to raw crawled corpora. CulturaX applied language identification, quality filtering, and deduplication to produce these cleaned subsets.

### 3.2 Pre-processing

RoBERTa employs the byte pair encoding (BPE) tokenizer originally introduced with GPT-2 (Radford et al., 2019), which processes raw text directly without requiring pre-tokenization or language-specific tools like Moses (Koehn et al., 2007). While this tokenizer was trained on English corpora, we followed the approach taken for GottBERT (Scheible et al., 2024) by training a dedicated Portuguese tokenizer. Using 40 GB of randomly sampled Portuguese corpus data, we created a 52k-token vocabulary optimized for the language. Although we did not explicitly measure the impact on file size or task performance for PortBERT, similar adaptations in Dutch (Delobelle et al., 2020) and German (Scheible et al., 2024) have demonstrated benefits in both respects. In our experience, a 40 GB sample is sufficient for the subword distribution to converge, and extending vocabulary training to the full corpus would add considerable overhead with little expected benefit.

### 3.3 Pre-training

Similar to GottBERT, we pre-trained the PortBERT$_{base}$ and PortBERT$_{large}$ models using the Fairseq framework. PortBERT$_{large}$ was trained on a 128-core TPUv4 pod (Jouppi et al., 2023), while PortBERT$_{base}$ was trained on a cluster of 8 NVIDIA A40 GPUs, using the same training corpus and identical optimization hyperparameters. Mixed-precision training (fp16) was disabled for the GPU setup and not supported by the TPU implementation used, ensuring that both models were trained entirely in full precision (fp32). This controlled setup enables a direct comparison of hardware-level training efficiency across compute architectures, without numerical precision optimizations acting as confounding factors. Both models were trained on Portuguese OSCAR data using the RoBERTa architecture. The PortBERT$_{base}$ model completed training in approximately 27 days (2,331,939 seconds), while PortBERT$_{large}$ required around 6.2 days (531,807 seconds). We used the standard RoBERTa pretraining schedule with 100k update steps, a batch size of 8k, a 10k-step warmup, and polynomial learning rate decay. The base model used a peak learning rate of 0.0004, and the large model 0.00015. As with GottBERT, we evaluated after each epoch and stored checkpoints throughout training. However, since the dataset size only permitted approximately four epochs, the final checkpoint coincided with the best-performing one.

### 3.4 Downstream Tasks

Based on the pre-trained BERT models, we fine-tuned several downstream tasks using the training scripts provided by Huggingface (Wolf et al., 2019). Hyperparameter optimization was performed via grid search, focusing on batch size and learning rate. Each task was trained for a maximum of 10 epochs, and the experiments were orchestrated using NNI (Microsoft, 2025) on NVIDIA A40 GPUs.

To assess model performance, each downstream task was fine-tuned 28 times using different combinations of batch sizes and learning rates. Since no separate test set was available, we selected the best-performing checkpoint based on validation set scores. The final performance figures reported for each model and task reflect the best result among these 28 validation-based runs. For comparison, we benchmarked our models against eleven other Portuguese language models.

We evaluated the models on ExtraGLUE (Santos et al., 2025b), a Portuguese adaptation of the GLUE benchmark. This suite consists of selected tasks from GLUE and SuperGLUE that were automatically translated into Portuguese, enabling language-specific assessment and ensuring that model performance reflects capabilities in the target language context.

To account for varying input lengths across tasks, we configured the maximum input sequence length individually per task based on the maximum observed input lengths after tokenization across all evaluated models: 192 tokens for MRPC and WNLI, 320 tokens for STS-B, and 512 tokens for RTE. This ensured full coverage of the datasets while avoiding unnecessary padding and memory overhead.

**STS-B** The Semantic Textual Similarity Benchmark (STS-B) task evaluates the model's ability to assess the semantic similarity between two sentences. Each sentence pair is assigned a similarity score ranging from 0 (completely dissimilar) to 5 (semantically equivalent). Following standard practice, we report the mean of Pearson and Spearman correlation coefficients between predicted and gold scores.

**RTE** The Recognizing Textual Entailment (RTE) task consists of binary classification, where the model must determine whether a given hypothesis logically follows from a provided premise. This task evaluates the model's capacity for inference and semantic reasoning.

**WNLI** The Winograd Natural Language Inference (WNLI) task is a coreference resolution challenge cast as binary entailment. It requires the model to resolve ambiguous pronouns and determine whether a hypothesis follows from a premise. Despite its small size and challenging structure, it is retained for completeness and consistency with GLUE-style benchmarks.

**MRPC** The Microsoft Research Paraphrase Corpus (MRPC) task is a binary classification problem where the model must decide whether two sentences are semantically equivalent. Evaluation is based on both accuracy and F1 score, reflecting the importance of both precision and recall in paraphrase detection.

### 3.5 Model Configurations and Properties

The number of parameters in BERT-like models varies significantly depending on their architecture (see Table 2). The base version of BERT, such as BERTimbau$_{\text{base}}$, has approximately 109 million parameters, while large versions like BERTimbau$_{\text{large}}$ expand to over 334 million. RoBERTa variants used in Portuguese NLP, such as RoBERTaCrawlPT$_{\text{base}}$ and RoBERTaLexPT$_{\text{base}}$, feature around 125 million parameters, comparable to PortBERT$_{\text{base}}$ (126M). The large PortBERT model increases this to 357 million, positioning it close to BERTimbau$_{\text{large}}$ while retaining RoBERTa's efficiency characteristics.

Multilingual models such as XLM-RoBERTa are designed for cross-lingual tasks, with the base version containing 278 million parameters and the large version 560 million. These parameter counts make them substantially larger than monolingual base models, but beneficial in zero-shot or cross-lingual scenarios (Eronen et al., 2023).

The AiBERTa and AlBERTina families offer diverse parameter ranges. All AiBERTa variants (regardless of source or domain configuration) have approximately 101 million parameters, with a smaller vocabulary size of 20,000. The AlBERTina models, in contrast, range from 138 million (100M variants) to over 1.5 billion parameters for the 1.5B variants, reflecting a significant increase in capacity and vocabulary size (up to 128,100 tokens). These models serve different use cases depending on the required balance between compute and performance.

Finally, EuroBERT models span from 210 million parameters in the 210M variant to over 2.1 billion in the 2.1B variant. They provide a scalable foundation for multilingual or European-centric tasks, emphasizing both vocabulary coverage and model depth.

Table 2: The size of the vocabulary and the size of the parameters are shown for the model types used in this study. This table does not show other design differences of the models. Values were extracted using Huggingface's transformers library. Models are sorted by number of parameters.

| Model | Vocab Size | #Params |
|---|---|---|
| roBERTa PT | 32000 | 68090880 |
| AiBERTa | 20000 | 101401344 |
| $\text{BERTimbau}_{\text{base}}$ | 29794 | 108923136 |
| $\text{RoBERTaLexPT}_{\text{base}}$ | 50265 | 124645632 |
| $\text{RoBERTaCrawlPT}_{\text{base}}$ | 50265 | 124645632 |
| $\text{PortBERT}_{\text{base}}$ | 52009 | 125985024 |
| AlBERTina 100M PTPT | 50265 | 138601728 |
| AlBERTina 100M PTBR | 50265 | 138601728 |
| EuroBERT 210m | 128256 | 211766016 |
| XLM $\text{RoBERTa}_{\text{base}}$ | 250002 | 278043648 |
| $\text{BERTimbau}_{\text{large}}$ | 29794 | 334396416 |
| $\text{PortBERT}_{\text{large}}$ | 52009 | 357145600 |
| XLM $\text{RoBERTa}_{\text{large}}$ | 250002 | 559890432 |
| EuroBERT 610m | 128256 | 607874688 |

# 4 Results

## 4.1 Downstream task evaluation

Table 3 presents the downstream evaluation results of all Portuguese language models across four ExtraGLUE tasks: STS-B, RTE, WNLI, and MRPC. We report task-specific metrics: Spearman and Pearson correlations for STS-B, accuracy for RTE and WNLI, and both accuracy and F1 for MRPC, alongside the average performance (AVG) across all tasks.

Among the base-sized models, $\text{RoBERTaLexPT}_{\text{base}}$ achieves the highest overall score with an AVG of 80.63, showing strong results particularly in MRPC accuracy (89.46) and F1 (92.34). Close behind is $\text{PortBERT}_{\text{base}}$, with an AVG of 80.57, outperforming all others in WNLI accuracy (60.56, tied with XLM-R) and ranking second in STS-B with a Spearman score of 87.39 and Pearson of 87.65. $\text{BERTimbau}_{\text{base}}$ shows the best performance in STS-B (88.5 mean), but underperforms slightly in WNLI, holding it back from overall top placement.

$\text{RoBERTaCrawlPT}_{\text{base}}$ and EuroBERT 210m also demonstrate robust overall performance, particularly in RTE and MRPC, with AVG scores above 79.0. Meanwhile, XLM $\text{RoBERTa}_{\text{base}}$ shows competitive results in WNLI (60.56) and MRPC F1 (91.32), though its STS-B score slightly lags behind the top contenders. Legacy models like roBERTa PT perform significantly worse, especially on semantic similarity tasks, confirming the impact of more recent training strategies and data sources.

In the large model category, XLM $\text{RoBERTa}_{\text{large}}$ emerges as the strongest overall model with an AVG of 84.01. It leads all others in STS-B (90.14 mean) and achieves the highest RTE score (82.31), although it underperforms in WNLI. EuroBERT 610m follows closely with an AVG of 83.44, showing outstanding performance in MRPC (94.2 F1, 91.91 accuracy) and the second-best RTE result (78.34).

$\text{PortBERT}_{\text{large}}$ achieves a solid overall score of 82.26, slightly ahead of $\text{BERTimbau}_{\text{large}}$ (82.23). While $\text{BERTimbau}_{\text{large}}$ does not dominate any single task, $\text{PortBERT}_{\text{large}}$ exhibits the highest WNLI accuracy (61.97). $\text{BERTimbau}_{\text{large}}$ stands out with strong STS-B scores (89.5 mean) and competitive MRPC metrics.

Overall, the results validate the effectiveness of the PortBERT models, with both the base and large variants frequently ranking among the top-performing models across tasks. The base model outperforms many existing Portuguese models on average, while the large model achieves results close to the best multilingual transformers. This indicates their robustness and applicability to a range of semantic and inference tasks in Portuguese.

## 4.2 Performance vs. Efficiency

To complement accuracy-based comparisons, we also assess model efficiency in terms of training and inference throughput (see Figure 1). Among the base models, several exhibit a favorable balance between performance and efficiency. Notably, roBERTa PT achieves the highest training throughput (62.1 samples/sec) and inference speed (112.7 samples/sec), but its task performance lags significantly behind all competitors, suggesting that efficiency alone is insufficient without adequate pretraining quality. In contrast, $\text{PortBERT}_{\text{base}}$ and $\text{RoBERTaCrawlPT}_{\text{base}}$ both demonstrate strong downstream performance (AVG: 80.57 and 80.48, respectively) while maintaining competitive training throughput around 25–26 samples/sec and inference throughput above 65 samples/sec. $\text{BERTimbau}_{\text{base}}$ similarly offers

| Model | STS-B (Similarity) | | | RTE | WNLI | MRPC | | AVG |
|---|---|---|---|---|---|---|---|---|
| | **Spearman** | **Pearson** | **Mean** | **Acc** | **Acc** | **Acc** | **F1** | |
| BERTimbau$_{\text{large}}$ | <u>89.4</u> | <u>89.61</u> | <u>89.5</u> | 75.45 | <u>59.15</u> | 88.24 | 91.55 | 82.23 |
| EuroBERT 610m | 88.46 | 88.59 | 88.52 | <u>78.34</u> | <u>59.15</u> | **91.91** | **94.2** | <u>83.44</u> |
| XLM RoBERTa$_{\text{large}}$ | **90.0** | **90.27** | **90.14** | **82.31** | 57.75 | <u>90.44</u> | <u>93.31</u> | **84.01** |
| PortBERT$_{\text{large}}$ | 88.53 | 88.68 | 88.6 | 72.56 | **61.97** | 89.46 | 92.39 | 82.26 |
| AiBERTa | 83.56 | 83.73 | 83.65 | 64.98 | 56.34 | 82.11 | 86.99 | 76.29 |
| AlBERTina 100M PTBR | 85.97 | 85.99 | 85.98 | 68.59 | 56.34 | 85.78 | 89.82 | 78.75 |
| AlBERTina 100M PTPT | 86.52 | 86.51 | 86.52 | 70.04 | 56.34 | 85.05 | 89.57 | 79.01 |
| BERTimbau$_{\text{base}}$ | **88.39** | **88.6** | **88.5** | <u>70.4</u> | 56.34 | 87.25 | 90.97 | 80.32 |
| EuroBERT 210m | 86.54 | 86.62 | 86.58 | 65.7 | 57.75 | 87.25 | 91.0 | 79.14 |
| RoBERTaCrawlPT$_{\text{base}}$ | 87.34 | 87.45 | 87.39 | **72.56** | 56.34 | <u>87.99</u> | 91.2 | 80.48 |
| RoBERTaLexPT$_{\text{base}}$ | 86.68 | 86.86 | 86.77 | 69.31 | <u>59.15</u> | **89.46** | **92.34** | **80.63** |
| XLM RoBERTa$_{\text{base}}$ | 85.75 | 86.09 | 85.92 | 68.23 | **60.56** | 87.75 | <u>91.32</u> | 79.95 |
| PortBERT$_{\text{base}}$ | <u>87.39</u> | <u>87.65</u> | <u>87.52</u> | 68.95 | **60.56** | 87.75 | 91.13 | <u>80.57</u> |
| roBERTa PT | 48.06 | 48.51 | 48.29 | 56.68 | <u>59.15</u> | 72.06 | 81.79 | 61.04 |

Table 3: Evaluation results in %. STSB is reported with Spearman, Pearson, and their mean. RTE and WNLI are classification accuracy. MRPC includes accuracy and F1. The AVG score averages the six metrics: STSB Spearman, STSB Pearson, RTE Acc, WNLI Acc, MRPC Acc, MRPC F1. Bold = best, underlined = second-best per model size. Based on best epoch from 28 runs for max 10 epochs. The AVG score is computed as the unweighted mean across six metrics: STS-B Spearman, STS-B Pearson, RTE accuracy, WNLI accuracy, MRPC accuracy, and MRPC F1.

a good trade-off with strong performance (AVG: 80.32) and respectable throughput, making these three the most efficient base models when balancing quality and compute.

The large models generally exhibit higher downstream performance but at a considerable computational cost. XLM RoBERTa$_{\text{large}}$ leads in task performance (AVG: 84.01) and inference throughput (47.4 samples/sec) compared to its large-model peers. However, its training throughput is relatively low (14.9 samples/sec), indicating longer training durations. PortBERT$_{\text{large}}$ achieves an attractive efficiency-performance trade-off, with an AVG of 82.26 while maintaining higher training and inference throughput (23.3 and 70.7 samples/sec, respectively), positioning it as the most throughput-efficient large model while still achieving competitive accuracy. Meanwhile, EuroBERT-610M delivers strong performance (AVG: 83.44) but with lower throughput metrics, reflecting its high computational demands. These results suggest that while large models provide superior accuracy, the efficiency gap between well-optimized base and large models like PortBERT is narrowing. Full runtime statistics are reported in Appendix C.

## 5 Discussion

### 5.1 Efficiency and Accuracy Trade-offs

PortBERT demonstrates that efficient, monolingual transformer models remain a valuable asset in the evolving landscape of Portuguese NLP. While large multilingual encoders like XLM-RoBERTa or EuroBERT-610M offer strong performance, their high computational demands restrict practical deployment, particularly in latency-sensitive or resource-constrained settings. In contrast, PortBERT delivers competitive downstream task results while maintaining generally higher throughput compared to other strong Portuguese baselines, both during training and inference.

As shown in our efficiency analysis (Section 4.2), PortBERT$_{\text{base}}$ stands out for its balanced trade-off between accuracy and efficiency, ranking among the top performers in its class. PortBERT$_{\text{large}}$ narrows the performance gap to state-of-the-art models like XLM RoBERTa$_{\text{large}}$, while maintaining superior throughput and lower hardware demands. Our focus with PortBERT was on cost-efficient pretraining for Portuguese specifically, where zero-shot transfer is not required. In this sense, PortBERT complements large multilingual encoders such as XLM-RoBERTa by offering a more efficient option for monolingual applications.

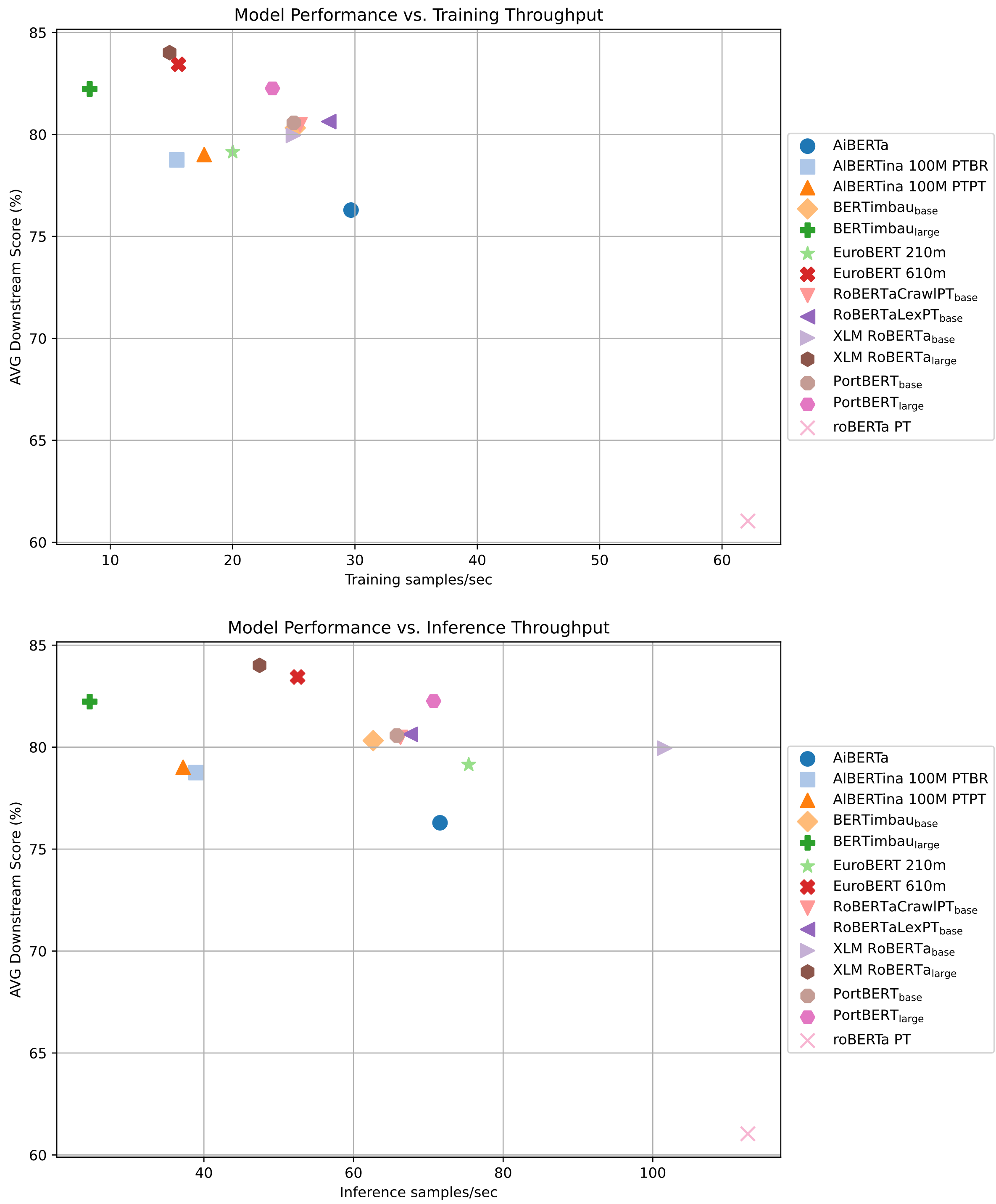



Figure 1: Performance–throughput trade-off across models. The top plot shows the relationship between average downstream score (AVG) and training throughput (samples/sec), while the bottom plot presents the same metric against inference throughput. This comparison highlights which models offer the best balance between effectiveness and computational efficiency during training and inference.

The performance differences between PortBERT and large multilingual encoders such as XLM-RoBERTa$_{large}$ are not solely attributable to the amount of training data. They also reflect architectural and training differences, including the substantially larger parameter count of XLM-RoBERTa (560M vs. 357M for PortBERT$_{large}$), its much larger multilingual vocabulary (250k vs. 52k tokens), and the use of a massive multilingual corpus (2.5TB multilingual vs. 456GB of Portuguese).

In addition to hardware throughput, PortBERT models also demonstrate strong parameter efficiency. PortBERT$_{\text{base}}$ (126M parameters) achieves higher average performance than larger models such as XLM-RoBERTa$_{\text{base}}$ (278M) and EuroBERT-210M (212M), despite having less than half their parameter count. PortBERT$_{\text{large}}$ (357M) achieves results close to XLM-RoBERTa$_{\text{large}}$ (560M) and EuroBERT-610M (608M), highlighting the impact of targeted, monolingual pretraining on recent Portuguese corpora. This makes PortBERT a compelling choice in scenarios where both accuracy and model size matter.

### 5.2 Training Setup and Hardware Comparisons

Beyond per-job throughput, the total pretraining time differed substantially between hardware setups. PortBERT$_{\text{base}}$, trained on 8 NVIDIA A40 GPUs, required approximately 27 days to complete 100k update steps. In contrast, PortBERT$_{\text{large}}$, trained on a TPUv4 128 pod, completed training in just over 6 days. Both models used the same batch size, corpus, and optimizer settings in full precision (fp32), allowing for a clean comparison of training performance across hardware platforms. Using GottBERT's pretraining durations as a reference, we estimate that PortBERT$_{\text{base}}$ would have taken around 1.3 days to train on comparable TPU infrastructure. This illustrates the advantage of modern TPUs for large-scale training, particularly when time is a critical factor. However, TPU-specific constraints, including limited memory flexibility and less mature tooling for PyTorch and custom workflows, can limit development. In addition, the lack of local TPU hardware forces developers to rely on cloud platforms, slowing iteration and complicating debugging.

Efficiency comparisons must also consider hardware configuration. Due to memory constraints, EuroBERT-610M and partly XLM RoBERTa$_{\text{large}}$ were trained without parallel jobs (i.e., one job per GPU), whereas PortBERT and other models used multiple parallel training jobs per GPU to maximize utilization. This difference in hardware allocation might have impacted the observed throughput and training durations, potentially skewing efficiency comparisons in this regard.

### 5.3 Positioning Among Existing Models

Recent large-scale efforts such as EuroBERT (Boizard et al., 2025) illustrate the scale-performance frontier in multilingual modeling. EuroBERT training consumed over 200,000 GPU hours across MI250X and MI300A clusters and leveraged cutting-edge optimization techniques such as FlashAttention (Dao, 2023). While such models raise the performance ceiling, they also require infrastructure that is out of reach for many academic or industry teams. In contrast, PortBERT was trained on commodity hardware using open-source tools, offering a transparent and efficient alternative that lowers the entry barrier for building high-quality models in any languages.

To our knowledge, PortBERT is the first RoBERTa-style Portuguese model trained on recent deduplicated and filtered corpora from CulturaX (mC4) and OSCAR23, using a fully transparent and reproducible fairseq pipeline. This positions it as a strong alternative to more resource-intensive systems, particularly for researchers and practitioners seeking open, efficient solutions.

Although decoder-only models such as GPT variants dominate general-purpose NLP, they are often unsuitable for sentence-level classification tasks due to their autoregressive nature. Encoder-based models like PortBERT offer lower inference latency and better fit for downstream classification, especially under real-world constraints.

### 5.4 Architectural Constraints and Training Stability

We deliberately retained the standard RoBERTa encoder architecture. Our goal was not only to establish a strong monolingual baseline, but also to enable a fair comparison of computational costs with GottBERT, which was trained on a comparable TPU setup. Introducing architectural modifications such as sparse or FlashAttention would have shifted the baseline and made this comparison meaningless.

Like GeistBERT (Scheible-Schmitt and Frei, 2025), PortBERT prioritizes practical usability over raw scale. Although it does not achieve top performance on every benchmark, it remains consistently strong across tasks, making it a compelling option in the accuracy-efficiency trade-off. PortBERT could also be adapted for longer inputs using architectures such as Longformer (Beltagy et al., 2020) or Nyströmformer (Xiong et al., 2021), though at the cost of increased training complexity.

During pretraining, we did not apply WWM, as stable support for it was missing in the fairseq

TPU implementation. As with GottBERT, we encountered TPU-specific constraints: the lack of dynamic memory allocation required processing the corpus as a continuous token stream, deviating from RoBERTa's dynamic sentence-sampling strategy. We were also constrained to 32-bit precision due to unstable 16-bit support in fairseq's TPU implementation, increasing memory use and runtime. To ensure stability under these conditions, we used conservative learning rates. For comparability, we deliberately applied the same pre-processing and training constraints to the GPU-based base model, even though the GPU setup would have supported dynamic sampling and mixed precision.

### 5.5 Final Remarks

Ultimately, PortBERT is a step toward sustainable and accessible language modeling for Portuguese. It illustrates that thoughtful model design, combined with optimized pretraining and recent corpora, can yield strong models without relying on large-scale infrastructure. Future work may explore quantized or distilled versions for mobile deployment and domain-specific continued pretraining to further expand applicability or even continue pretraining with a more diverse corpus using WWM similar to Scheible-Schmitt and Frei (2025).

## 6 Conclusion

We presented PortBERT, a family of RoBERTa-based language models for Portuguese, pre-trained on recent large-scale corpora (mC4 and OSCAR23). While not state-of-the-art on all benchmarks, PortBERT models achieve strong downstream performance and demonstrate notable efficiency in training and inference. To support reproducibility and downstream adoption, we release both Huggingface-compatible models and fairseq checkpoints. These resources enable further pretraining, fine-tuning, or adaptation for longer contexts and domain-specific tasks. PortBERT offers an efficient and accessible foundation for Portuguese NLP.

## Acknowledgments

We gratefully acknowledge the support of Google's TensorFlow Research Cloud (TFRC) for providing access to Cloud TPUs, which enabled efficient pretraining of PortBERT$_{\text{large}}$. We also thank Nora Limbourg, our Google Cloud Customer Engineer, for her valuable technical assistance and coordination throughout the project.

R.S. would also like to thank Bruno & Suzi, as well as all members and friends of the Best Spot Azores Diving Center, including Alberto & Simona, Arturo, João & Claudia, Maëlle & Elias, Maria, Oliver, Paula, Raquel, Ruben, Sara and Vasco. Their kindness, presence, and community spirit provided strength and stability in a time of personal challenge. It is always a pleasure to dive with us.

## Limitations

This work has several limitations. First, although we used deduplicated and filtered corpora from CulturaX (mC4 and OSCAR23), we did not apply deduplication across all possible data sources or levels of granularity. Residual duplication or noise may therefore remain in the training data.

Second, PortBERT was trained exclusively on web-based Portuguese text, without explicit control for dialectal variation (e.g., Brazilian vs. European Portuguese) or domain-specific content. As a result, the model's performance on underrepresented dialects or specialized registers (e.g., legal, medical, or informal language) may be suboptimal without further fine-tuning.

Third, while we aimed for stable and reproducible training configurations across both GPU and TPU platforms, we opted for conservative learning rates and default precision settings to ensure stability, particularly on TPUs where dynamic memory allocation and mixed precision remain limited in fairseq. We did not explore extensive hyperparameter tuning in regard of the peak learning rate and did not apply WWM, which could potentially yield further gains.

Fourth, we did not include a detailed error analysis of model predictions. While such an analysis could provide additional insights into systematic failure modes, our focus in this work was on efficiency and establishing strong baselines for Portuguese NLP.

Lastly, our evaluation is focused on the ExtraGLUE benchmark. While this provides a useful proxy for general NLP performance in Portuguese, it does not capture the full range of downstream tasks or real-world deployment settings. Moreover, ExtraGLUE does not offer a held-out test set with a submission server, which limits the ability to conduct blind evaluations and compare models in a standardized manner.

## Ethical Considerations

As with any large-scale language model, PortBERT is susceptible to inheriting and reproducing biases present in its training data. While we apply deduplication techniques to reduce noise and redundancy, deeper societal, cultural, and representational biases may persist. This is particularly relevant for downstream applications in sensitive domains such as healthcare, education, or public administration, where biased outputs could reinforce inequality or misinformation.

Training on large-scale web-based corpora also introduces privacy concerns. Although the dataset is filtered and preprocessed, models may inadvertently memorize and surface sensitive or personal information. Careful handling is necessary when deploying PortBERT in real-world applications, especially those involving user data or decision-making contexts.

Finally, despite efforts to balance performance and efficiency, pretraining transformer models on GPUs and TPUs consumes substantial computational resources. The associated energy usage and environmental impact underline the importance of developing sustainable training practices and promoting model reuse.

## References


Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2024. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Nicolas Boizard, Hippolyte Gisserot-Boukhlef, Duarte M. Alves, André Martins, Ayoub Hammal, Caio Corro, Céline Hudelot, Emmanuel Malherbe, Etienne Malaboeuf, Fanny Jourdan, Gabriel Hautreux, João Alves, Kevin El-Haddad, Manuel Faysse, Maxime Peyrard, Nuno M. Guerreiro, Patrick Fernandes, Ricardo Rei, and Pierre Colombo. 2025. Eurobert: Scaling multilingual encoders for european languages.

Branden Chan. 2020. XLM-RoBERTa: The multilingual alternative for non-english NLP. Library Catalog: towardsdatascience.com.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. *arXiv:2001.06286 [cs]*. ArXiv: 2001.06286.

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.

Juuso Eronen, Michal Ptaszynski, and Fumito Masui. 2023. Zero-shot cross-lingual transfer language selection using linguistic similarity. *Information Processing & Management*, 60(3):103250.

Eduardo A. S. Garcia, Nadia F. F. Silva, Felipe Siqueira, Hidelberg O. Albuquerque, Juliana R. S. Gomes, Ellen Souza, and Eliomar A. Lima. 2024. RoBERTaLexPT: A legal RoBERTa model pretrained with deduplication for Portuguese. In *Proceedings of the 16th International Conference on Computational Processing of Portuguese*, pages 374–383, Santiago de Compostela, Galicia/Spain. Association for Computational Linguistics.

Tim Jansen, Yangling Tong, Victoria Zevallos, and Pedro Ortiz Suarez. 2022. Perplexed by quality: A perplexity-based method for adult and harmful content detection in multilingual heterogeneous web data.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Norman P. Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Cliff Young, Xiang Zhou, Zongwei Zhou, and David Patterson. 2023. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

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.

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:1907.11692 [cs]*. ArXiv: 1907.11692.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osae Osae Dade, Wenhao Yu, Lucas Krauß, Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Muñoz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2024. Starcoder 2 and the stack v2: The next generation.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe.

Microsoft. 2025. Neural network intelligence. `https://github.com/microsoft/nni/`. Accessed: 2025-05-01.

Nuno Miquelina, Paulo Quaresma, and Vítor Beires Nogueira. 2022. Generating a european portuguese bert based model using content from arquivo.pt archive. In *Intelligent Data Engineering and Automated Learning – IDEAL 2022*, pages 280–288, Cham. Springer International Publishing.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

João Rodrigues, Luís Gomes, João Silva, António Branco, Rodrigo Santos, Henrique Lopes Cardoso, and Tomás Osório. 2023. *Advancing Neural Encoding of Portuguese with Transformer Albertina PT-**, page 441–453. Springer Nature Switzerland.

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

Daniel Santos, Nuno Miquelina, Daniela Schmidt, Paulo Quaresma, and Vítor Beires Nogueira. 2025a. Performance evaluation of nlp models for european portuguese: Multi-gpu/multi-node configurations and optimization techniques. In *Algorithms and Architectures for Parallel Processing*, pages 298–314, Singapore. Springer Nature Singapore.

Daniel Santos, Nuno Miquelina, Daniela Schmidt, Paulo Quaresma, and Vítor Beires Nogueira. 2025b. Performance Evaluation of NLP Models for European Portuguese: Multi-GPU/Multi-node Configurations and Optimization Techniques. In *Algorithms and Architectures for Parallel Processing*, pages 298–314, Singapore. Springer Nature.

Rodrigo Santos, João Rodrigues, António Branco, and Rui Vaz. 2021. Neural text categorization with transformers for learning portuguese as a second language. In *Progress in Artificial Intelligence*, pages 715–726, Cham. Springer International Publishing.

Raphael Scheible, Johann Frei, Fabian Thomczyk, Henry He, Patric Tippmann, Jochen Knaus, Victor Jaravine, Frank Kramer, and Martin Boeker. 2024. GottBERT: a pure German language model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21237–21250, Miami, Florida, USA. Association for Computational Linguistics.

Raphael Scheible-Schmitt and Johann Frei. 2025. Geistbert: Breathing life into german nlp.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. Bertimbau: Pretrained bert models for brazilian portuguese. In *Intelligent Systems*, pages 403–417, Cham. Springer International Publishing.

Jorge A Wagner Filho, Rodrigo Wilkens, Marco Idiart, and Aline Villavicencio. 2018. The brwac corpus: A new open resource for brazilian portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. 2021. Nyströmformer: A nyström-based algorithm for approximating self-attention.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

## A Parameters

The parameter space for our grid search is listed in Table 4. In addition, Table 5 shows the parameters of the best models (selection based on validation set) of the respective tasks. We include these details to support reproducibility of our downstream results.

| Parameter | Values |
|---|---|
| Learning Rate | 7e-5, 5e-5, 2e-5, 1e-5, 7e-6, 5e-6, 1e-6 |
| Batch Size | 16, 32, 48, 64 |
| Epochs | 10 |

Table 4: Hyperparameters used in the grid search of the downstream tasks.

## B Perplexity

During pretraining, model perplexity was tracked on a test set after each optimization step and on a validation set at every checkpoint (see Figure 2). The models exhibited a plateau in their perplexity curves, brief for the base models, but more prolonged for the large ones. Some training curves also showed temporary spikes, which may appear as divergence if not interpreted with context. Across both models, convergence occurred gradually and stabilized by around 30k steps. In contrast, the validation perplexity decreased steadily across both models without showing pronounced plateaus, stabilizing at lower values by the end of training. This results from the limited number of validation checkpoints (three intermediate epochs and a final checkpoint at 100k steps), which yield a coarser view of the learning dynamics.

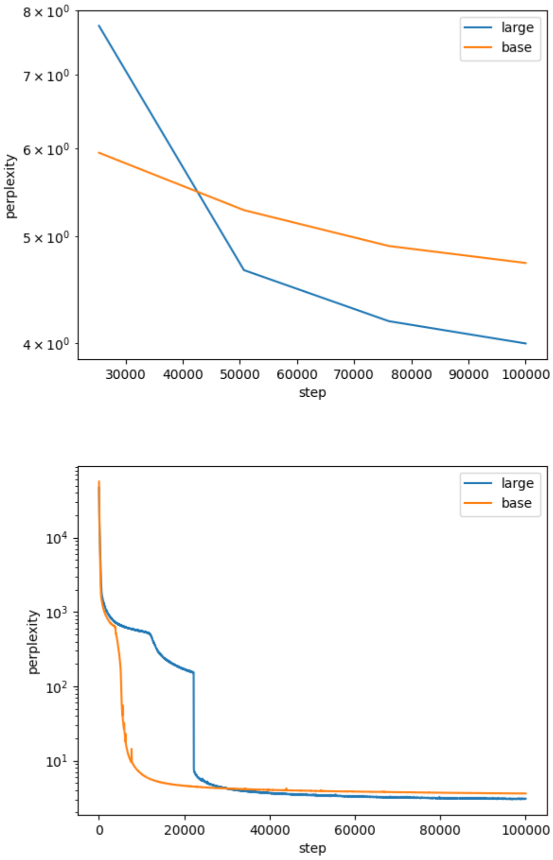



Figure 2: Perplexity of the PortBERT models. Top based on a validation at the checkpoints. Bottom based on the validation of each optimization cycle during the training.

## C Efficiency Measurements

Tables 6 and 7 report detailed runtime statistics for all models and tasks. Table 6 provides a task-level breakdown of training and inference times, while Table 7 compares model-level efficiency metrics, including throughput and per-epoch timing. All models were fine-tuned using Huggingface Transformers (v4.52.3) on NVIDIA A40 GPUs.

| Task | Training Time | Inference Time |
|---|---|---|
| MRPC | 157:04 | 00:38 |
| RTE | 241:46 | 00:57 |
| STSB | 314:24 | 02:25 |
| WNLI | 25:30 | 00:08 |

Table 6: Computation time in hours and minutes for the downstream tasks, summing up to 1549 hours and 29 minutes, which corresponds to approximately 64.6 days of GPU usage.

| Model | STS-B | | RTE | | WNLI | | MRPC | |
|---|---|---|---|---|---|---|---|---|
| | BS | LR | BS | LR | BS | LR | BS | LR |
| EuroBERT 210m | 16 | 7 E-06 | 64 | 2 E-05 | 32 | 2 E-05 | 32 | 7 E-06 |
| XLM RoBERTa$_{large}$ | 64 | 2 E-05 | 32 | 1 E-05 | 16 | 7 E-05 | 64 | 2 E-05 |
| AlBERTina 100M PTPT | 64 | 7 E-05 | 32 | 2 E-05 | 16 | 2 E-05 | 48 | 5 E-05 |
| AlBERTina 100M PTBR | 64 | 5 E-05 | 16 | 1 E-05 | 32 | 1 E-06 | 48 | 7 E-05 |
| AiBERTa | 32 | 2 E-05 | 32 | 1 E-05 | 32 | 7 E-05 | 32 | 5 E-05 |
| EuroBERT 610m | 16 | 1 E-05 | 16 | 7 E-06 | 32 | 1 E-05 | 16 | 5 E-06 |
| XLM RoBERTa$_{base}$ | 16 | 1 E-05 | 32 | 2 E-05 | 64 | 2 E-05 | 16 | 2 E-05 |
| roBERTa PT | 32 | 7 E-05 | 32 | 1 E-05 | 48 | 5 E-06 | 32 | 7 E-05 |
| RoBERTaCrawlPT$_{base}$ | 48 | 7 E-05 | 64 | 7 E-05 | 48 | 1 E-06 | 48 | 7 E-05 |
| BERTimbau$_{large}$ | 32 | 2 E-05 | 16 | 7 E-05 | 32 | 7 E-06 | 16 | 5 E-05 |
| BERTimbau$_{base}$ | 48 | 5 E-05 | 16 | 1 E-05 | 48 | 7 E-05 | 32 | 7 E-05 |
| PortBERT$_{base}$ | 48 | 7 E-05 | 16 | 1 E-05 | 16 | 1 E-06 | 64 | 1 E-05 |
| RoBERTaLexPT$_{base}$ | 48 | 5 E-05 | 48 | 5 E-05 | 32 | 7 E-06 | 64 | 2 E-05 |
| PortBERT$_{large}$ | 16 | 2 E-05 | 16 | 7 E-06 | 32 | 7 E-06 | 16 | 7 E-06 |

Table 5: Hyperparameters of the best downstream task models for each task and pre-trained model. BS refers to batch size, and LR denotes the learning rate.

| **Model** | **Train Time (s)** | **Train/s** | **Time/Epoch (s)** | **Eval Time (s)** | **Eval/s** |
|---|---|---|---|---|---|
| AiBERTa$_{2000M}$ | 1306.47 | 29.68 | 142.39 | 7.24 | 71.56 |
| AlBERTina$_{PTBR}$ | 2906.82 | 15.44 | 309.08 | 15.85 | 38.92 |
| AlBERTina$_{PTPT}$ | 2800.95 | 17.68 | 300.35 | 17.54 | 37.22 |
| BERTimbau$_{base}$ | 1499.94 | 25.12 | 152.88 | 9.15 | 62.63 |
| BERTimbau$_{large}$ | 4406.49 | 8.32 | 484.90 | 21.44 | 24.73 |
| EuroBERT$_{210M}$ | 1777.84 | 20.01 | 181.90 | 6.60 | 75.40 |
| EuroBERT$_{610M}$ | 2498.58 | 15.58 | 254.26 | 12.80 | 52.52 |
| RoBERTaCrawlPT$_{base}$ | 1682.64 | 25.51 | 171.76 | 9.15 | 66.28 |
| RoBERTaLexPT$_{base}$ | 1457.99 | 27.86 | 149.42 | 8.84 | 67.58 |
| XLM-RoBERTa$_{base}$ | 1440.55 | 24.97 | 152.49 | 4.86 | 101.59 |
| XLM-RoBERTa$_{large}$ | 2139.34 | 14.85 | 233.65 | 10.46 | 47.44 |
| PortBERT$_{base}$ | 1524.59 | 25.00 | 160.29 | 8.96 | 65.79 |
| PortBERT$_{large}$ | 2389.63 | 23.26 | 264.74 | 15.09 | 70.71 |
| roBERTa PT | 635.46 | 62.11 | 79.02 | 4.82 | 112.71 |

Table 7: Training and inference efficiency of all evaluated models. Metrics include total training time, training samples per second, average time per epoch, total evaluation time, and evaluation throughput.

# Quality Matters: Measuring the Effect of Human-Annotated Translation Quality on English-Slovak Machine Translation

**Matúš Kleštinec**
Constantine the Philosopher University in Nitra
Tr. A. Hlinku 1, 949 01 Nitra, Slovakia
matus.klestinec@ukf.sk

**Daša Munková**
Constantine the Philosopher University in Nitra
Tr. A.Hlinku 1, 949 01 Nitra, Slovakia
dmunkova@ukf.sk

## Abstract

This study investigates the influence of human-annotated translation quality on the performance of machine translation (MT) models for a low-resource language pair—English to Slovak. We collected and categorized 287 student translations from a national competition, annotated by expert translators into three quality levels. Using the mT5-large model, we trained six neural MT models: three on the full dataset without validation splitting, and three using training/validation splits. The models were evaluated using a suite of automatic metrics (BLEU, METEOR, chrF, COMET, BLEURT, and TER), with TER serving as the validity criterion. Statistical analyses revealed that data quality had no significant effect when training without validation, but did have a significant impact under fine-tuning conditions ($p < 0.05$). Our results suggest that fine-tuning with combination with validation splitting increases the model's sensitivity to the quality of training data. While the overall effect size is modest, the findings underscore the importance of high-quality, annotated corpora and modern training strategies for improving MT in low-resource languages.

## 1 Introduction

Machine translation (MT) refers to the use of algorithms and machine learning models to translate texts from one natural language into another (Keary, 2023). Modern MT systems increasingly rely on artificial neural networks, which can autonomously learn to perform translation with high accuracy - often achieving levels of accuracy comparable to those of human translators (Young, 2024). Building a high-quality MT model typically requires access to large volumes of training data. Although neural approaches have reached state-of-the-art performance in MT, they suffer from the high cost of acquiring large-scale parallel corpora (Wang et al., 2021).

A neural MT model $\theta$ translates a source sentence $x$ into a target sentence $y$. Using a parallel training corpus $C$, the model $\theta$ is trained by minimizing the negative log-likelihood loss. The encoder-decoder structure (based on recurrent neural networks, convolutional neural networks or transformer) is commonly employed in neural MT, where the encoder transforms the source sentence into a sequence of hidden representations and the decoder generates target words based on these representations and the previously generated target words (Wang et al., 2021). For high-resource language pairs such as English-French, data availability is less problematic, as substantial parallel corpora have been compiled over time. However, the requirement for large amounts of parallel data is often unrealistic for many of the 7000+ languages spoken worldwide, which presents a major challenge for low-resource languages (Ranathunga et al., 2023). The low-resource problem may stem either from a language itself is low-resourced (underrepresented) or from specific domains lack sufficient data (Hedderich et al., 2021).

In the case of the Slovak language, the limited availability of text data categorizes it as a low-resource language (Do et al., 2014). Such languages are underrepresented in digital spaces compared to high-resource languages, making it difficult for speakers to use the advaced technologies in their daily lives - including effective neural MT systems (Tonja et al., 2023).

The research objective:



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 72–81, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-009

The aim of this study is to investigate how both the quality of parallel texts (fair, good, and excellent translations) and the distribution of the dataset (corpus) influence MT system performance, specifically the quality of neural MT output as measured by automatic evaluation metrics.

The structure of this study is as follows: Section 1 introduces the research problem, motivation, and contributions. Section 2 reviews related work on data quality in MT and prior studies on evaluation metrics. Section 3 describes the dataset, tokenization process, model setup, and evaluation metrics. Section 4 presents the experimental results, including statistical analyses of models trained on both the full dataset and split dataset. Section 6 concludes the study and outlines directions for future work.

## 2 Related work

A recent case study demonstrated that carefully targeted data collection can significantly improve MT performance in a low-resource language pair (Hasan et al. 2020). Data is arguably the most critical factor in modeling (developing) translation systems (Haddow et al., 2022). When applying data-driven MT to a specific language pair, the initial step involves assessing available data resources and identifying effective strategies for collecting additional data. In the context of low-resource MT, Haddow et al. (2022) classify research approaches into four main categories: searching existing data sources, web-crawling for parallel data, data creation, and test data development. In our research, we focus on creating a new parallel dataset comprising student translations from English into Slovak.

Several researchers have explored the use of multiple references in MT. Wu et al. (2024) measured semantic similarity among reference translations and categorized them into different training subsets based on their degree of variation. They fine-tuned two pre-trained large language models - LLAMA-2-7B and mT5-large - using datasets containing multiple references. Their results showed that using source texts with semantic similarity scores between 0.45 and 1.0 led to better performance than unfiltered datasets. Similarly, Zouhar et al. (2021) investigated how the quality and quantity of reference translations affect the reliability of automatic MT evaluation metrics. They found that low-quality or overly diverse references may distort metric scores, whereas carefully selected multiple references enhance evaluation robustness. Our study builds on these findings by combining both perspectives: we employ multiple reference translations per source sentence while accounting for diversity in human-annotated translation quality. Unlike prior studies that primarily focused on semantic variation, we examine how quantity and quality of human-annotated translations influences MT model training and quality of MT outputs.

## 3 Methodology

### 3.1 Data collection and pre-processing

The texts used in this study were obtained from the Young Translator public competition, which is open to high school students interested in translation. A total of 287 student translations were included in this study, most of which were translations of literary texts. Two professional translators - both university lecturers in translation and interpreting - evaluated the translations and classified them into three quality categories: 1 – fair translation, 2 – good translation, and 3 – excellent translation. Since the collected translations were available only in printed form, several pre-processing steps were required before training.

The following pre-processing steps were applied:

- Optical character recognition (OCR)
- Text editing for alignment
- Alignment of English and Slovak texts
- Additional text editing prior tokenization and training
- Tokenization

#### Optical character recognition

Because the original documents were available only as scanned PDFs, it was necessary to convert them into machine-readable text. This was achieved using the Tesseract OCR library (Tesseract OCR, 2025). Although the student translations (essays) were typewritten, many contained handwritten annotations—often in black or colored ink—as part of the evaluation process. In cases where colored pens were used, color filtering was applied to improve OCR accuracy. After recognition, the output was stored in txt format for further processing.

**Text editing for alignment**

The OCR output required extensive cleaning. Common issues included misrecognized characters, extra punctuation marks (e.g., quotation marks), incorrect spacing (e.g., multiple spaces), and line breaks not corresponding to sentence boundaries. All empty lines were removed to prevent alignment errors. Additionally, the texts were anonymized to remove any personally identifiable information.

**Alignment of English and Slovak sentences**

After cleaning, the English source texts and Slovak translations were aligned. Each English sentence corresponded to multiple Slovak translations (a 1-to-*n* alignment), reflecting the multiple student versions. To facilitate semantic alignment, we employed LaBSE (Language-agnostic BERT Sentence Embedding), a model trained on more than 100 languages, including English and Slovak (Feng et al. 2020). A similarity threshold of 0.6 was applied.

The aligned data were merged into two larger txt files - one for English and one for Slovak - structured for training purposes. Each model requires one text file in English and one corresponding text file in Slovak. Duplicate sentence pairs were removed, and the dataset was randomly shuffled. Since the number of sentence pairs varied across the three quality categories, the sets for scores 2 and 3 were downsampled to match the smallest set (score 1), ensuring balanced training data and avoiding bias in model evaluation (Table 1). For English and Slovak, the number of words was (54,827 EN | 43,354 SK) for model_1, (54,056 EN | 42,849 SK) for model_2, and (55,514 EN | 42,717 SK) for model_3. Number of tokens was (100,841 EN | 97,284 SK) for model model_1, (101,698 EN | 96,707 SK) for model_2, and (101,802 | 96,406) for model_3.

| | **Model_1** | **Model_2** | **Model_3** |
|---|---|---|---|
| **Slovak sentences** | 3130 | 3130 | 3130 |
| **English sentences** | 3130 | 3130 | 3130 |

Table 1: Number of sentences for each model

**Tokenization**

For tokenization and training of MT models, we utilized the pre-trained *mT5-large* model. This model is based on the transformer architecture and was trained on a multilingual dataset containing sentences from 101 languages, including Slovak (Xue et al., 2020). The *mT5-large* model was used to tokenize both the English and Slovak texts in preparation for training. We selected this model because it covers the English-Slovak language pair and offers a balance between model capacity and training efficiency.

Although the *mT5* model includes Slovak in its pre-training data, fine-tuning on domain-specific datasets is still necessary to achieve optimal performance. In this study, we trained separate models for each quality category, resulting in a total of six models:

1) Three models trained on the full dataset for each category (fair, good, and excellent translations).

2) Three models trained on a split dataset for each category (fair, good, and excellent translations).

All training was conducted on Google Colab using an NVIDIA A100 GPU.

## 3.2 Models trained on the full dataset

In the first experiment, we trained three models using the entire dataset. Each model corresponded to one of the three quality categories - fair, good, and excellent translations.

The training parameters for these models are summarized in Table 2:

| **Hyperparameters** | **Values** |
|---|---|
| Per_device_train_batch_size | 4 |
| Num_train_epochs | 3 |
| Learning_rate | 1e-4 |
| fp16 | False |

Table 2: Hyperparameters for training

After training, three MT models were obtained. Their performance was evaluated using a reference file containing all unique English–Slovak sentences, which had been excluded from the training data to ensure a fair and unbiased evaluation.

## 3.3 Models with data split

The key difference between the initial three models and the subsequent three lies in the data split strategy. For these latter models, the dataset was randomly divided into 90% for training and 10% for validation. The 10% validation set was used to fine-tune the models during training. The hyperparameters employed for training all three fine-tuned models are listed in Table 3.

| Hyperparameters | Value |
|---|---|
| Per_device_train_batch_size | 4 |
| Num_train_epochs | 5 |
| Learning_rate | 1e-4 |
| fp16 | False |
| eval_strategy | steps |
| eval_steps | 500 |

Table 3: Hyperparameters for training

### 3.4 Evaluation metrics

The trained models were evaluated using a range of automatic metrics: BLEU, METEOR, COMET chrF, TER, and BLEURT.

**BLEU** (BiLingual Evaluation Understudy) is a precision-based metric that evaluates MT output by comparing n-grams in the hypothesis (MT output) with those in one or more reference translations. It does not consider word order beyond matching n-grams and tends to reward exact matches. A higher BLEU score indicates closer overlap with the reference and therefore better translation quality (Papineni et al., 2002).

**METEOR** (Metric for Evaluation of Translation with Explicit Ordering) is a metric that aligns words and phrases between the hypothesis and reference translations using synonyms, stemming, and paraphrasing. It calculates scores based on unigram precision, recall, and F-score, which are combined via a weighted harmonic mean. Score ranges from 0 (poor translation) to 1 (perfect translation) (Banerjee et al. 2005).

**COMET** is a neural framework that considers both source and reference translations. Trained on human judgment data, it predicts sentence-level quality score. This study employed several versions, including wmt20-comet-da, wmt21-comet-da, wmt21-comet-qe-da and wmt22-comet-da. Metric wmt22-comet-da integrates quality estimation techniques using OK/BAD tags from human-annotated datasets and combines multiple models via hyperparameter optimization to produce a single quality score (Rei et al. 2020, Rei et al. 2022). Scores typically range from 0 (poor quality) to 1 (high quality).

**BLEURT** (Bilingual Evaluation Understudy with Representations from Transformers) is a regression-based evaluation metric built on BERT. Fine-tuned on human ratings of translation quality, it captures subtle semantic differences between translations. BLEURT scores generally range from 0 to 1, though values may occasionally exceed this range due to the nature of the regression output (Sellam et al., 2020).

**chrF** is a character n-gram F-score metric that evaluates translation quality at the character level rather than the word level. This approach is particularly effective for morphologically rich languages or those with flexible word order. It computes F-scores over character n-grams (e.g., 6-grams), combining precision and recall into a single score, with higher values indicating better translation quality (Popović, 2015).

**TER** (Translation Edit Rate) measures the number of edits (insertions, deletions, substitutions, and shifts) required to transform the hypothesis into the reference translation. Lower TER score indicates higher translation quality, as fewer edits are needed (Snover et al., 2006).

## 4 Results

To facilitate interpretation and comparison of MT model performance, the evaluation metrics were grouped according to their scale and underlying evaluation strategy. Three metric groups were defined:

- Group 1 (within-group factor: Metric1*): Includes BLEU, METEOR, chrF, wmt22-comet-da and wmt21-comet-qe-da. These metrics primarily assess surface-level or structural similarity between hypothesis and reference translations.
- Group 2 (within-group factor: Metric2*): Includes BLEURT, wmt20-comet-da, and wmt21-comet-da. These metrics capture deeper semantic similarity, often leveraging pre-trained language representations and human rating data.
- Group 3: TER, treated as a separate metric due to its nature as an error-based measure, serving as a validity criterion for the accuracy metrics.

We hypothesize that statistically significant differences will exist among the examined metrics, between within-group metric (Metric1*/Metric2*) and the translation quality levels of the training data (between-group factor: quality levels 1–3).

TER as the only metric explicitly measuring edit distance/error rate, is used as a benchmark validity measure to evaluate the reliability and consistency of the other metrics.

### 4.1 Models trained on full dataset

To assess the assumption of homogeneity of variances across the independent variable (quality levels 1, 2, and 3) a nonparametric Levene's test was used. The results were non-significant (Table 4), indicating that the assumption of equal variances between independent groups was not violated.

| | MS Effect | MS Error | F | p |
|---|---|---|---|---|
| bleu | 0.019 | 0.018 | 1.038 | 0.355 |
| meteor | 0.004 | 0.013 | 0.323 | 0.724 |
| chrf | 0.002 | 0.011 | 0.183 | 0.833 |
| wmt22-comet-da | 0.003 | 0.006 | 0.447 | 0.640 |

Table 4: Levene' test for homogeneity of variances

However, the assumption of sphericity - which concerns the equality of variances of the differences between all combinations of dependent metrics in Metric1 (BLEU, METEOR, chrF, and wmt22-comet-da) - was violated (Table 5).

| | W | Chi-Sqr. | df | p |
|---|---|---|---|---|
| Metric1 | 0.496 | 396.360 | 5 | 0.0000 |

Table 5: Mauchley sphericity test

To preserve statistical power and ensure the validity of the analysis, adjusted univariate tests for repeated measures were applied. These tests evaluated the effects of translation quality and its interaction with evaluation metric (metric1 × quality) on translation performance (Table 6).

| | Epsilon | Adj. df1 | Adj. df2 | Adj. p |
|---|---|---|---|---|
| Metric1 | 0.705 | 2.116 | 1199.867 | 0.0000 |
| Metric1 x quality | 0.705 | 4.232 | 1199.867 | 0.4542 |

Table 6: Adjusted (G-G) univariate tests for repeated measure

Statistically significant differences ($p < 0.05$) were observed only among the evaluation metrics themselves (Table 6). The effect of between-group factor (quality level) did not have a statistically significant on the evaluation outcomes ($p > 0.05$), indicating that the quality categories (1, 2, and 3) did not significantly influence scores across the metrics.

The results of the multilevel comparison (Table 7) further clarify the relative behavior of individual metrics. Specifically, a statistically significant difference was found between the BLEU metric and remaining metrics, whereas no statistically significant difference was observed between meteor and chrf metrics. Even when considering the quality levels (Table 7), no statistically significant differences were found between the individual quality categories (1, 2, and 3) with respect to the metrics included in Metric1 (BLEU, METEOR, chrF, and wmt22-comet-da).

| Quality | Metric1 | mean | 1 | 2 | 3 |
|---|---|---|---|---|---|
| 3 | bleu | 0.255 | | **** | |
| 1 | bleu | 0.282 | | **** | |
| 2 | bleu | 0.299 | | **** | |
| 3 | meteor | 0.593 | **** | | |
| 3 | chrf | 0.598 | **** | | |
| 1 | meteor | 0.606 | **** | | |
| 1 | chrf | 0.616 | **** | | |
| 2 | meteor | 0.628 | **** | | |
| 2 | chrf | 0.635 | **** | | |
| 1 | wmt22-comet-da | 0.820 | | | **** |
| 3 | wmt22-comet-da | 0.825 | | | **** |
| 2 | wmt22-comet-da | 0.837 | | | **** |
| Note: **** - p > 0.05, homogeneous group | | | | | |

Table 7: Multi-stage comparison

We applied the same analytical procedure to Group Metric2, taking into account deviations from the assumption of normality.

Statistically significant differences were observed only among the evaluation metrics within-group Metric2 ($p = 0.000$). The effect of the between-group factor (translation quality level) on evaluation scores was not statistically significant ($p = 0.552$), indicating that the assigned quality categories did not influence the metric scores in this group. Statistically significant differences were found between all three metrics in this group ($p < 0.05$). When incorporating the translation quality factor, no significant interaction effects based on translation quality were observed (Table 8).

| quality | metric2 | Mean | 1 | 2 | 3 |
|---|---|---|---|---|---|
| 3 | wmt21-comet-da | 0.050 | **** | | |
| 1 | wmt21-comet-da | 0.056 | **** | | |
| 2 | wmt21-comet-da | 0.071 | **** | | |
| 3 | bleurt | 0.096 | **** | **** | |
| 1 | bleurt | 0.134 | **** | **** | |
| 2 | bleurt | 0.179 | | **** | |
| 3 | wmt20-comet-da | 0.625 | | | **** |
| 1 | wmt20-comet-da | 0.638 | | | **** |
| 2 | wmt20-comet-da | 0.696 | | | **** |
| Note: **** - p > 0.05, homogeneous group | | | | | |

Table 8: Multi-stage comparisons

| metric2 | MS Effect | MS Error | F | p |
|---|---|---|---|---|
| bleurt | 0.019 | 0.055 | 0.346 | 0.7077 |
| wmt20-comet-da | 0.053 | 0.110 | 0.487 | 0.6147 |
| wmt21-comet-da | 0.000 | 0.008 | 0.001 | 0.9994 |

Table 9: Levene' Test for Homogeneity of Variances

| | W | Chi-Sqr. | df | p |
|---|---|---|---|---|
| metric2 | 0.728 | 179.615 | 2 | 0.0000 |

Table 10: Mauchley Sphericity Test

| | Epsilon | Adj. df1 | Adj. df2 | Adj. p |
|---|---|---|---|---|
| metric2 | 0.786 | 1.572 | 891.567 | 0.0000 |
| metric2 x quality | 0.786 | 3.145 | 891.567 | 0.5521 |

Table 11: Adjusted (G-G) Univariate Tests for Repeated Measure

Similar to the first group of metrics, deviations from normality were identified for the second group of metrics. Based on the results of the nonparametric Levene's test (Table 9), we conclude that the assumption of equality of variances between independent samples (quality: 1, 2, and 3) is not violated. In the case of dependent samples (metric2: bleurt, wmt20-comet-da, wmt21-comet-da), the sphericity condition of the covariance matrix was violated (Table 10). In order not to reduce the power of the statistical tests, we use adjusted univariate tests for repeated measures (Table 11) to assess the quality of the translation as a function of the interaction of the within-group and between-group factors (metric2 x quality).

Statistically significant differences were observed only among the metrics themselves ($p < 0.05$), while the between-group factor, translation quality, did not have a significant effect on evaluation outcomes ($p > 0.05$) (Table 8). A multilevel comparison (Table 8) indicates that the wmt21-comet-da metric is statistically the most rigorous metric ($p < 0.05$), whereas wmt20-comet-da is statistically the least rigorous ($p < 0.05$). Statistically significant differences were found between all three metrics ($p < 0.05$).

The reliability analysis of the MT assessment procedure indicates that the selected set of evaluation metrics - BLEU, METEOR, chrF, BLEURT, wmt22-comet-da, wmt20-comet-da, and wmt21-comet-da - demonstrates acceptable internal consistency (*Cronbach's* $\alpha > 0.6$), suggesting that the metrics collectively form a coherent measurement construct (*Average inter-item corr.* > 0.5).

The MT evaluation procedure explains nearly 70% of the variability in MT error rate (Table 12). Based on the validity analysis (Table 12), the procedure demonstrates acceptable criterion validity. The TER metric, which directly represents MT error rate, was employed as the validity criterion (Munk et al., 2018), confirming that the combined use of BLEU, METEOR, chrF, BLEURT, wmt22-comet-da, wmt20-comet-da, and wmt21-comet-da provides a valid estimation of translation accuracy.

| | **Summary for scale** |
|---|---|
| Multiple R | 0.830 |
| Multiple R2 | 0.689 |
| F(7,562) | 178.169 |
| p | 0.0000 |

Table 12: Validity analysis

## 4.2 Models with data split

As with the first three models, evaluation of the split-data models was performed using the same reference file. Due to deviations from normality and differences in the range of the evaluated scores, the metrics were again divided into three groups, following the same grouping strategy as in the first experiment.

We hypothesize that statistically significant differences will exist between within-group metric (Metric1*/Metric2*) and the translation quality levels of the training data (between-group factor: quality levels 1–3).

As in the previous analysis, the TER metric (ter_ref.) was employed as the validity criterion, since it directly reflects the MT error rate. As in the previous analysis, we observed a violation of the sphericity assumption for the covariance matrix, as indicated by the Mauchly's Test of Sphericity ($p < 0.05$), which pertains to the use of repeated (dependent) measures (metric1*: bleu_ref., meteor_ref., chrf_ref., wmt22-comet-da, wmt21-comet-qe-da).

We applied adjusted univariate tests for repeated measures to evaluate translation quality as a function of the interaction between within-group (metric1)* and between-group (quality level) factors. The results indicated statistically significant differences among the evaluated metrics ($p = 0.000$), as well as a significant effect of translation quality on the evaluation outcomes ($p = 0.004$).

Multilevel comparisons (Table 13) further confirmed statistically significant differences among all metrics ($p < 0.05$). Additionally, a significant effect of translation quality was observed across nearly all metrics, except wmt21-comet-qe-da, for which the effect was not statistically significant ($p > 0.05$).

| **quality** | **metric1*** | **Mean** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|---|---|
| 3 | wmt21-comet-qe-da | 0.106 | **** | | | | | | |
| 2 | wmt21-comet-qe-da | 0.107 | **** | | | | | | |
| 1 | wmt21-comet-qe-da | 0.107 | **** | | | | | | |
| 3 | bleu_ref. | 0.218 | | **** | | | | | |
| 1 | bleu_ref. | 0.271 | | | **** | | | | |
| 2 | bleu_ref. | 0.283 | | | **** | | | | |
| 3 | meteor_ref. | 0.546 | | | | **** | | | |
| 3 | chrf_ref. | 0.555 | | | | **** | | | |
| 1 | meteor_ref. | 0.596 | | | | | **** | | |
| 2 | meteor_ref. | 0.611 | | | | | **** | | |
| 1 | chrf_ref. | 0.619 | | | | | **** | | |
| 2 | chrf_ref. | 0.627 | | | | | **** | | |
| 3 | wmt22-comet-da | 0.801 | | | | | | **** | |
| 1 | wmt22-comet-da | 0.821 | | | | | | **** | **** |
| 2 | wmt22-comet-da | 0.842 | | | | | | | **** |
| Note: **** - p > 0.05, homogeneous group | | | | | | | | | |

Table 13: Multi-stage comparisons

| **quality** | **metric2*** | **Mean** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|---|
| 8 | bleurt_ref. | 0.019 | **** | | | |
| 8 | wmt21-comet-da | 0.024 | **** | | | |
| 6 | wmt21-comet-da | 0.062 | **** | | | |
| 7 | wmt21-comet-da | 0.072 | **** | | | |
| 7 | bleurt_ref. | 0.172 | | **** | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 6 | bleurt_ref. | 0.174 | | **** | | |
| 8 | wmt20-comet-da | 0.54 | | | | **** |
| 6 | wmt20-comet-da | 0.65 | | | **** | |
| 7 | wmt20-comet-da | 0.708 | | | **** | |
| Note: **** - p > 0.05, homogeneous group | | | | | | |

Table 14: Multi-stage comparisons

For the second group of metrics (metric2*: bleurt_ref., wmt20-comet-da, wmt21-comet-qe-da), the sphericity assumption was also violated (Mauchley Sphericity Test: $p < 0.05$). In order not to reduce the power of the statistical tests, we use modified univariate tests for repeated measures to assess the quality of the translation as a function of the within-group and between-group interaction (metric2* x quality) (Table 15).

| | Epsilon | Adj. df1 | Adj. df2 | Adj. p |
|---|---|---|---|---|
| metric2* | 0.787 | 1.575 | 892.795 | 0.0000 |
| metric2* x quality | 0.787 | 3.149 | 892.795 | 0.0130 |

Table 15: Adjusted (G-G) univariate tests for repeated measure

Statistically significant differences (Table 15) were again demonstrated between the metrics themselves ($p < 0.05$), and the effect of translation quality was likewise significant ($p < 0.05$).

When including translation quality as a factor in the multilevel comparison (Table 14), a statistically significant influence of quality level was confirmed for almost all metrics, with the exception of wmt21-comet-da ($p > 0.05$).

| | Summary for scale |
|---|---|
| Multiple R | 0.890 |
| Multiple R2 | 0.792 |
| F(7,562) | 267.144 |
| p | 0.0000 |

Table 16: Validity analysis

The MT evaluation procedure explains nearly 80% of the variability in the MT error rate (Table 16). Based on the results of the validity analysis (Table 16), we conclude that the procedure demonstrates acceptable criterion validity. The TER metric was employed as the validity criterion (Munk et al., 2018), confirming that the combined use of BLEU, METEOR, chrF, BLEURT, wmt22-comet-da, wmt20-comet-da, and wmt21-comet-da provides a valid estimation of translation accuracy.

## 5 Conclusion

The study demonstrates that the quality of annotated training data influences the performance of neural MT systems for the English–Slovak language pair. However, the extent of this effect depends strongly on the training strategy. When models were trained on the full dataset without validation splitting, translation quality level showed no significant impact on performance ($p > 0.05$). In contrast, when the dataset was split into training and validation subsets, translation quality level significantly affected the evaluation metrics ($p < 0.05$). This suggests that fine-tuning with held-out validation data increases the model's sensitivity to training data quality.

Despite minor deviations and variations across individual metrics, the overall evaluation procedure explains a significant proportion of the variance in translation error rates. These findings indicate that for low-resource languages such as Slovak, enhancing the quality of human-annotated parallel corpora can lead to measurable gains in MT performance - particularly when modern training strategies like fine-tuning on held-out validation sets are employed. Nonetheless, the effect size remains relatively small, and further improvements may require not only higher-quality data, but also larger and more diverse training corpora.

## Acknowledgments

This work was supported by the Slovak Research and Development Agency under the Contract no. APVV-23-0554.

## References

Anthony Young, 2024. Exploring *Machine Translation: Output Quality, Learner Reflection, Teacher Detection*. In *Thailand TESOL Conference Proceedings 2024.*

Atnafu Lambebo Tonja, Olga Kolesnikova, Alexander Gelbukh and Grigori Sidorov. 2023. *Low-Resource Neural Machine Translation Improvement Using Source-Side Monolingual Data.* In *Applied Sciences.* 2023; 13(2):1201. https://doi.org/10.3390/app13021201

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl and Alexandra Birch; *Survey of Low-Resource Machine Translation*. In *Computational Linguistics 2022; 48 (3):* 673–732. https://doi.org/10.1162/coli_a_00446

Britannica (2025): *OCR*. https://www.britannica.com/technology/OCR

Cong-Thanh Do, Lori Lamel and Jean-Luc Gauvain 2014. *Speech-to-text development for Slovak, a low-resourced language*. In 4th Workshop on Spoken Language Technologies for Under-resourced Languages, pages 176-182.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan and Wei Wang 2020. *Language-agnostic BERT Sentence Embedding.* In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Association for Computational Linguistics. arXiv:2007.01852. Version 2

Hasan, Tahmid, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. *Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali-English machine translation.* In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP),* pages 2612–2623. https://doi.org/10.18653/v1/2020 .emnlp-main.207

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. *A Method for Automatic Evaluation of Machine Translation*. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics (ACL),* pages 311-318. Philadelphia, Pennsylvania. https://doi.org/10.3115/1073083.1073135

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua and Colin Raffel. *2020. mT5: A massively multilingual pre-trained text-to-text transformer*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* Association for Computational Linguistics, pages 483-498. arXiv:2010.11934

Maja Popović. 2015. *chrF: character n-gram F-score for automatic MT evaluation*. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. *A Study of Translation Edit Rate with Targeted Human Annotation.* In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Michael A. Hedderich, Lukas Lange, Heike Adel, Jannik Strötgen, and Dietrich Klakow. 2021. *A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios*. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics*: *Human Language Technologies*. 2545–2568.

Michal Munk, Daša Munková and Lubomír Benko. 2018. *Towards the use of entropy as a measure for the reliability of automatic MT evaluation metrics.* In *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology 34(5)* https://doi.org/10.3233/JIFS-169505

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur and André F. T. Martins. 2022. *COMET-22: Unbabel-IST 2022 Submission for the Metrics Shared Task*. In *Proceedings of the Seventh Conference on Machine Translation (WMT).* Association for Computational Linguistics, pages 578-585. Abu Dhabi, United Arab Emirates.

Ricardo Rei, Craig Stewart, Ana C Farinha, Alon Lavie. 2020. *COMET: A Neural Framework for MT Evaluation*. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 2685-2702.

Rui Wang, Xu Tan, Renqian Luo, Tao Qin and Tie-Yan Liu. 2021. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence Survey Track*. Pages 4636-4643.

Satanjeev Banerjee and Alon Lavie. 2023. *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments.* In *Proceedings of the ACL Workshop on Instrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Association for Computational Linguistics, pages 65-72.

Si Wu, John Wieting and David A. Smith. 2024. *Multiple References with Meaningful Variatons Improve Literary Machine Translation*. arXiv:2412.18707

Surangika Ranathunga, En-Shiun Annie Lee, Marjana Prifti Skenduli, Ravi Shekhar and Mehreen Alam, Rishemjit Kaur. 2021. *Neural machine translation for low-resource languages*: In *A survey." ACM Computing Surveys 55(11),* pages 1-37. https://doi.org/10.48550/arXiv.2106.15115

Tesseract OCR (2025): Tesseract-ocr/tesseract. Github. https://github.com/tesseract-ocr/tesseract

Tim Keary. 2023. *Machine Translation (MT)*. https://www.techopedia.com/definition/machine-translation-mt

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. *BLEURT: Learning Robust Metrics for Text Generation.* In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Vilém Zouhar, Ondřej Bojar. 2024. *Quality and Qunatity of Machine Translation References for Automatic Metrics.* In *Proceedings of the Fourth Workshop on Human Evaluation of NLP Systems (HumEval) @ LREC-COLING 2024*, pages 1-11. arXiv:2401.01283. Version 4

# Spatio-Temporal Mechanism in Multilingual Sentiment Analysis

**Adarsh Singh Jadon**
ABV-IIITM / India
adarshjadon5@gmail.com

**Vivek Tiwari**
ABV-IIITM / India
vivektiwari@iiitm.ac.in

**Chittaranjan Swain**
ABV-IIITM / India
cswain@iiitm.ac.in

**Deepak Kumar Dewangan**
ABV-IIITM / India
deepakd@iiitm.ac.in

## Abstract

This study investigated the effectiveness of various models in deep learning in performing sentiment analysis on code-mixed Hinglish text, a hybrid language widely used in digital communication. Hinglish presents unique challenges due to its informal nature, frequent code-switching, and complex linguistic structure. This research leverages datasets from the HinGE, SemEval-2020 Task 9 & E-Commerce Reviews Datasets competition and employs models such as RNN (LSTM), BERT-LSTM, CNN, and a proposed BiLSTM model with Data Augmentation. The study's primary objective is to develop a robust sentiment analysis framework that accurately classifies sentiment in Hinglish text. The BiLSTM model demonstrated superior performance when trained and tested on 3 different datasets. This model outperformed existing approaches. The results highlight the proposed model's capability to handle the nuances of Hinglish in a more generalized manner, including its informal and code-mixed nature, more effectively than traditional models. The model also snags for future developments like data bias, interpretability of the model, scalability.

## 1 Introduction

During the last 10 years the worldwide expansion of social media with digital communication systems triggered an exceptional linguistic transformation leading to the emergence of multilingual hybrid expressions. Code-mixed languages have become prevalent markers of culture and globalization since they combine multiple languages in one communication context. Hinglish exists as the perfect example of a mixed language formed by combining Hindi and English that serves as a common communication method throughout India and throughout South Asia. However, Recurrent neural networks and their variant using long short-term memory units exhibit exceptional capability in identifying sequential dependencies in Hinglish text which enables them to process unstructured code-mixed sentences with diverse lengths. New opportunities for code-mixed language Sentiment Analysis emerged from the development of BERT (Bidirectional Encoder Representations from Transformers) and its transformer-based models & cross attention networks. The contextual embeddings produced by BERT-derived models like mBERT and IndicBERT obtain information from both local words and global linguistic patterns for the efficient disambiguation of polysemous words and hybrid phrases(He and Abisado, 2024; Hu et al., 2024; Li et al., 2024).However, the pathway faces multiple significant challenges that need to be resolved. The sociolinguistic diversity of Hinglish requires language frameworks to adapt through frameworks which understand continuous language development. The Hinglish language encompasses three levels of diversity that include geographical dialects together with differences between generational groups and technical jargon specific to platforms(Joshi et al., 2025). To counter this problem, this work presents Bidirectional LSTM(BiLSTM) model with Dense layers, with various data augmentation techniques. As a result, we get the features of LSTM, & various Dense layers help the model to get the most prominent features. As we know that text data can be uneven, which can lead to class imbalance. This work also showcases how data augmentation can be a technique which can tackle the problem of class imbalance on various datasets. This model is tested on 3 famous datasets - HinGE, E-Commerce Reviews & SemEval dataset. Comparing the results from previously implemented models, we get a descent result in general, and concluded that this model can be used on any kind of textual data.

The format of the paper is as follows: Section 2



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 82–89, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-010

emphasizes the relevant research which is done in this area, highlighting their methodologies & the dataset used. After that, Section 3 covers the proposed methodology along with the dataset used. Consequently, Section 4 showcases the comparative analysis and the proposed model's performance. Finally, Section 5 shows a brief conclusion & future work that can be done in this area.

## 2 Literature Review

Many researchers have developed various approaches in order to get the best performance of the model, using various machine learning techniques and deep learning architectures, or hybrid approaches. Let us look a few of them.

The authors in (Narang et al., 2024) operate to enhance misleading information detection powers by merging sentiment analysis techniques with text feature extractions. Their aim centers around building an effective method to detect false news within the current fast-paced digital era. The research draws from three different datasets including Covid-19.The proposed method delivers substantial performance gains that reach 20% accuracy betterment for detecting 2 classes in LIAR and 30% betterment when classifying 6 classes.TextGT in (Yin and Zhong, 2024) represents a new Aspect Based Sentiment Analysis(ABSA) approach which incorporates a double-view graph Transformer model according to the authors. The method implements specialized GNN layers for text graphs alongside Transformer layers for sequences while connecting them for resolving over-smoothing problems. The authors developed a new edge feature-enhanced graph convolution algorithm named TextGINConv for performance enhancement.The authors in (Alfreihat et al., 2024) create Emoji Sentiment Lexicon (Emo-SL) as part of their research to enhance sentiment analysis for Arabic tweets. The main goal focuses on managing the difficulties which arise from informal Arabic text specifically because of morphological complexity alongside language dialect variations. Combination of Emoji-based aspects with ML methods achieve enhanced sentiment classification because of their hybrid approach. A total of 58,000 Arabic tweets enter the dataset because they incorporate emojis. The collected dataset gathers tweets from the Arabic Sentiment Twitter database for achieving balanced positive/negative sentiment distribution. The model achieves an F1 score of 89% from sentiment classification and 26.7% in emoji feature extraction.
The authors in (Bilal et al., 2024) have set a goal to enhance sentiment classification accuracy through deep sequential feature combination with Random Forest (RF) technique application. The experimental results show that the proposed model detected 99.631% correct responses from the dataset which surpassed five baseline algorithms substantially. The research in (Li and Chen, 2024) investigates public discussions that focus on Virtual Humans together with their technological advancement and virtual idol and streamer applications and corporate investment along with policy strategies.The analysis tracked emotional tendencies as part of sentiment analysis procedures. Statistical analysis shows user discussions focus mainly on technological advancements of VHs and yield positive user reactions at 87.10%.This authors in (Mahmud et al., 2024) created a benchmark dataset dedicated to analyze sentiment in Cricket social media contents written in Bangla whereas the text comes from low-resource settings. The main purpose is to build better sentiment analysis tools for the Bangla language through an emphasis on cricket analysis since this content category stands as a major interest for Bangladesh. The research division established two parts for the dataset: training at 80% and testing at 20% which enabled a reliable assessment of model performance.Researchers in (Liu et al., 2024) work to resolve Multimodal Sentiment Analysis (MSA) difficulties which appear when uncertain missing modalities exist. The research introduces MTMSA which represents a novel modality translation-based Multi-Modal Sentiment Analysis model that improves sentiment classification outcomes through the proper use of text and audio and visual data. Gradual monologue videos in the CMU-MOSI dataset contain 2,199 instances that receive emotional score values between -3 & +3 and IEMOCAP presents extensive multimodal information for sentiment analysis. The authors in (Alsemaree et al., 2024) focus on sentiment analysis (SA) of Arabic social media texts, specifically targeting customer perceptions in the coffee industry. The text employs two methods of feature extraction for sentiment classification accuracy: Term Frequency-Inverse Document Frequency (TF-IDF) and Minimum Redundancy Maximum Relevance (MRMR). The researchers apply four supervised learning algorithms: KNN, support vector machine, decision tree and random forest for their analy-

sis. The newly proposed method reached an exceptional accuracy threshold of 95.95% using hard voting and 94.51% using soft voting. The authors in (Low et al., 2024) focus on creating a machine learning process which identifies and categorizes sexual harassment instances within literary documents while overcoming human interpretation shortcomings. Evaluation results demonstrated that the proposed LSTM-GRU deep learning model obtained 75.8% accuracy in sexual harassment type classification with superior performance compared to other five models. The same model design implemented for sentiment classification achieved an accuracy level of 84.5%. The authors in (Ramzy and Ibrahim, 2024) studied Arabic COVID-19 mobile health (mHealth) application user satisfaction by analyzing user review sentiments. The analysis used manual annotation of a representative 8,220 reviews to guarantee accurate sentiment identification. Six different machine learning systems consisting of Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naive Bayes (NB) and Logistic Regression (LR), Random Forest (RF) and Artificial Neural Network (ANN) were used to evaluate review sentiment. The ANN model revealed the best performance because it reached 89% accuracy and 89% F1 score. The study in (Lossio-Ventura et al., 2024) explores methods to assist healthcare providers and researchers with applying sentiment analysis tools to health-related free-text survey data when dealing with COVID-19. The authors used multiple human raters to establish gold-standard labels for a portion of their datasets that functioned as evaluation criteria for various sentiment analysis methods. The performed analysis demonstrated ChatGPT surpassing other sentiment analysis tools while also reaching superior accuracy values and F-scores. The accuracy scores from ChatGPT surpassed OPT by 6% and its F-measure results exceeded those of OPT by 4% to 7% across all datasets.

## 3 Proposed Framework

This section presents the proposed method that has been implemented on 2 different datasets. We begin our discussion on the proposed method from data collection, then going towards data preprocessing. After that, we will delve into exploratory data analysis(EDA). Next discussion will be on data augmentation, and then going through data distribution. Finally, we will describe our proposed methodology.

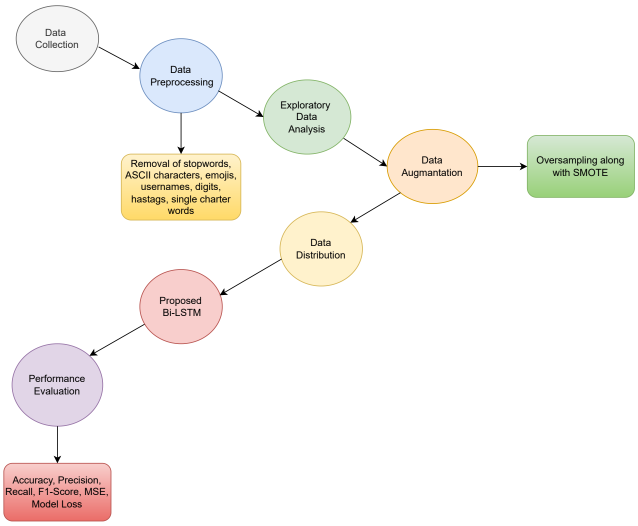



Figure 1: Proposed Workflow for the Sentiment Analysis using Bi-LSTM technique

### 3.1 Data Collection

We performed our experiment on the following datasets:

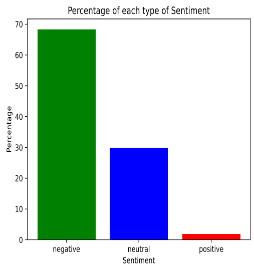
1. **HinGE Dataset:** The data originated from (Srivastava and Singh, 2021a), and futher used in (Jadon et al., 2024) provides 395, 2766, & 768 samples respectively for validation and training and testing purposes. The datasets present 3 sections namely ”English,” ”Hindi” and ”Hinglish” with synthetic Hinglish versions of Hindi & English.

2. **SemEval-2020 Task 9:** This dataset centers its analysis on the Twitter datasets of both Hinglish (Hindi–English) code-mixing along with Spanglish (Spanish–English). Comprising 19,000 tweets in Spanglish and 20,000 tweets in Hinglish contains sentiment classification and linguistic annotations for each tweet.

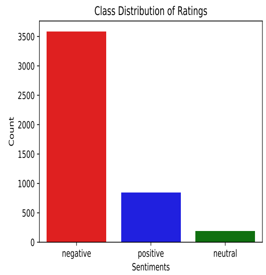
3. **E-Commerce Reviews Dataset:** This dataset has been generated by taking reviews from several E-Commerce platforms. It exhibits 10,000 reviews and their sentiment class distribution in ”positive”, ”neutral” or ”negative”.

### 3.2 Data Preprocessing

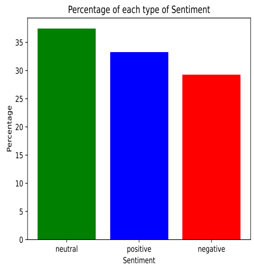
Initiating the process requires removing all additional symbols including #,,, %, and $.




(a) Dataset 1




(b) Dataset 2




(c) Dataset 3

Figure 2: Sentiment Class Distribution

Lowercase conversion of text represents one text cleaning operation while the removal of non-ASCII characters and numeric value stripping and punctuation mark elimination and hashtag symbol deletion and username blocking are additional procedures. Transaction tags ('RT') disappear while brief words and Hindi numerical digits are trimmed and URLs become indicated as "URL." The normalization process controls backslashes to improve consistency within the text data. Through their collective approach multiple cleaning techniques create a refined "processed text" column which ensures the Hinglish tweets undergo methodical cleaning procedures to become standardizable and free of noise until natural language processing can be employed.

### 3.3 Exploratory Data Analysis

The exploratory data analysis (EDA) approach serves as an advanced analytical stage to perform a complete evaluation of dataset quality together with its structural elements. Statistical and visual tools help researchers detect patterns and deviation points to determine suitable traits for future analysis during this method.

### 3.4 Data Augmentation

The data augmentation technique transforms training datasets to improve their overall quality together with their range of content while also making them more resistant to errors. The techniques create expanded datasets through processed variations of available data which maintain its original meaning or structural elements. We used 2 such techniques in our dataset - Oversampling along with SMOTE.

- **Oversampling:** Oversampling is conceptually simple: it duplicates existing samples from the minority class. Mathematically, this process involves selecting a sample $x_i$ from the minority class dataset $X_{\text{minority}}$ and adding it multiple times to the dataset. Let $X_{\text{minority}} = \{x_1, x_2, \ldots, x_m\}$ represent the minority class samples & $X_{\text{new}}$ represent the augmented dataset after oversampling. The procedure of oversampling can be expressed as:

$$X_{\text{new}} = X_{\text{minority}} \cup \bigcup_{j=1}^{k} \{x_i \mid x_i \in X_{\text{minority}}\}$$

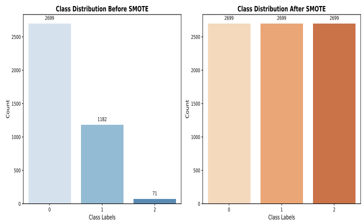
  Here $k$ is the number of times each sample $x_i$ is duplicated & the union operation ($\cup$) indicates that the original minority samples are combined with their duplicates. For example, if $X_{\text{minority}} = \{[1, 2], [3, 4]\}$ and $k = 2$, then:

$$X_{\text{new}} = \{[1, 2], [3, 4], [1, 2], [3, 4], [1, 2], [3, 4]\}$$

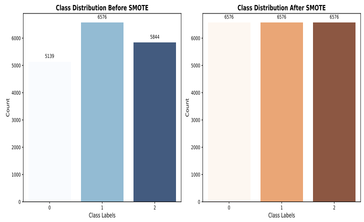
- **Synthetic Minority Over-sampling Technique(SMOTE):** SMOTE generates synthetic samples by interpolating between existing minority-class samples. It calculates new points along the line segment connecting a sample $x_i$ and one of its $k$-nearest neighbors $x_j$. Suppose $X_{\text{minority}} = \{x_1, x_2, \ldots, x_m\}$ represent the minority class samples, $\text{NN}_k(x_i)$ denote the set of $k$-nearest neighbors of $x_i$ in feature space & $\lambda \in [0, 1]$ represent a random interpolation factor. Then For each sample $x_i \in X_{\text{minority}}$ will randomly select a neighbor $x_j \in \text{NN}_k(x_i)$ and generate a synthetic sample $x_{\text{synth}}$ as:

$$x_{\text{synth}} = x_i + \lambda(x_j - x_i)$$

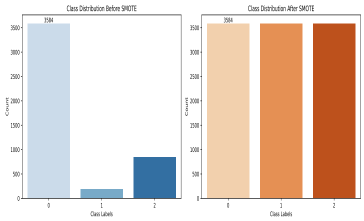
  where $x_i$ is the original sample, $x_j$ is the selected neighbor & $\lambda$ controls the position of the synthetic point along the line segment between $x_i$ and $x_j$. The augmented dataset $X_{\text{new}}$




(a) Dataset 1




(b) Dataset 2




(c) Dataset 3

Figure 3: EDA before and after Data Augmentation in all 3 datasets

is described as:

$$X_{\text{new}} = X_{\text{minority}} \cup \{x_{\text{synth}} \mid x_{\text{synth}} \text{ generated using the above formula}\}$$

### 3.5 Data Distribution

The application of an 70:20:10 data splitting ratio stands as a vital process step when dealing with sentiment analysis for Hinglish textual information ratio. The data was arranged into three subsets where validation data represents 20% of total data and test data comprises 10% of total data supplementary to training data containing the remaining 70%.

### 3.6 Proposed Model

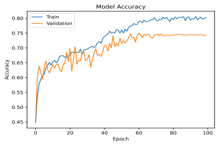
The implemented model utilizes a Bidirectional LSTM architecture that analyzes sequential data from forward and backward time sequences. The bidirectional method delivers the model access to contextual information flowing from past and future time points which boosts its ability to spot complex dependencies between sequence inputs. LSTMs enable strong performance in sentiment analysis and other applications with complex linguistic structures specifically in Hinglish language. The updation process governing the LSTM cell is shown in algorithm 1.

**Algorithm 1** LSTM Cell Updation

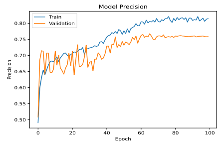
**Step 1: Data Preparation and Splitting**
**for** each sample $(x_i, y_i) \in \mathcal{D}$ **do**
    $x_i \leftarrow \texttt{LOAD}(x_i)$
    $x_i \leftarrow \texttt{PREPROCESS}(x_i)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_i, y_i)\}$
**end for**
$(\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}) \leftarrow \texttt{SPLIT}(\mathcal{D}, r = 0.2)$
**Step 2: Model Definition and Training**
Define model architecture: $\Phi \leftarrow \texttt{Sequential}(\text{FC}, \text{BN}, \text{Dropout}, \texttt{BiLSTM}, \text{FC})$
Initialize parameters $\theta$, optimizer Adam$(\eta = 0.001)$, and regularizers
**for** epoch = 1 to max_epochs **do**
    **for** each batch $B = \{x_j\}_{j=1}^{b} \in \mathcal{D}_{\text{train}}$ **do**
        $\hat{y}_j = \Phi(x_j)$
        $\mathcal{L} = \texttt{CrossEntropy}(\hat{y}, y) + \lambda \|\theta\|_2^2$
        $\nabla_\theta \mathcal{L} \leftarrow \texttt{BACKWARD}()$
        $\theta \leftarrow \texttt{OPTIMIZE}(\theta, \nabla_\theta \mathcal{L})$
    **end for**
    Evaluate validation loss $\mathcal{L}_{\text{val}}$
    **if** $\mathcal{L}_{\text{val}}$ not improving for `patience` epochs **then**
        `EarlyStopping` $\rightarrow$ break
    **else if** reduce_on_plateau condition met **then**
        $\eta \leftarrow \max(\eta \cdot 0.5, 10^{-6})$
    **end if**
**end for**
**Step 3: Save and Deploy Model**
$\theta^* \leftarrow \texttt{SAVE\_MODEL}(\theta)$
$\Phi^* \leftarrow \texttt{LOAD\_MODEL}(\theta^*)$

## 4 Results & Discussion

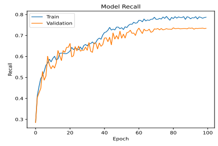
The use of Bidirectional LSTMs enhances the model's ability to capture both short-term and long-term dependencies in sequential data. This capability is crucial for analyzing the nuanced structure of Hinglish text, where context plays a vital role in determining sentiment. Additionally, the incorporation of regularization techniques such as dropout and L2 regularization ensures robust feature extraction and prevents overfitting, further improving the model's generalization capabilities. The model is compared on the basis



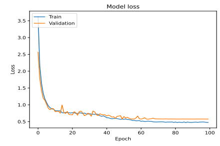
(a) Accuracy



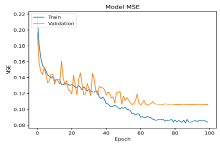
(b) Precision



(c) Recall

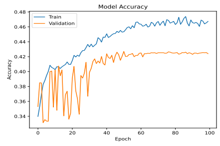


(d) Loss

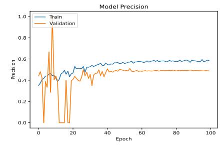


(e) MSE

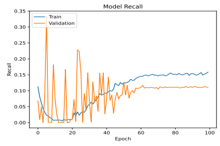
Figure 4: Learning Curve Analysis for the proposed Bi-LSTM model against Dataset 1(HinGE)



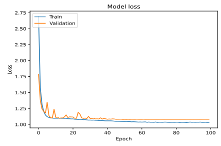
(a) Accuracy



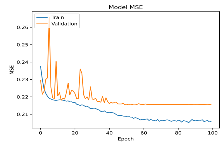
(b) Precision



(c) Recall



(d) Loss



(e) MSE

Figure 5: Learning Curve Analysis for the proposed Bi-LSTM model against Dataset 2(SemEval)

of the following metrics(Bala Das et al., 2023; Das et al., 2025):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

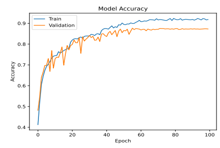
$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

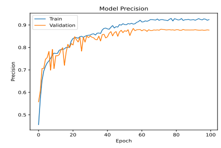
$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

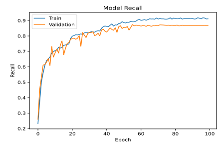
When we compared our model with the previously implemented models, the model was able to achieve the performance of the models on different datasets. As shown in Table 1, it gives us the 2nd highest performance in dataset 1. If we refer to Table 2, it again perfroms 2nd best performance in dataset 2. If we look at Table 3, it comes out to be the 5th best model among all the algorithms, which is in dataset 3.
In generalization, we can say that the model will perform as same as the previously implemented models, but it will generation of Model loss as shown in fig 4, 5, 6 respectively, and MSE will be low as compared to other models. In some cases, it will also outperform other models.



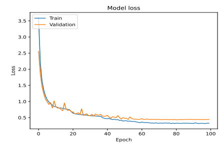
(a) Accuracy



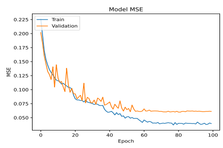
(b) Precision



(c) Recall



(d) Loss



(e) MSE

Figure 6: Learning Curve Analysis for the proposed Bi-LSTM model against Dataset 3(E-Commerce reviews)

## 5 Conclusion & Future Work

This paper implements a Bidirectional LSTM (BiLSTM) network as an efficient approach to resolve linguistic challenges existing in the combined Hindi & English language. The model stands out in detecting sentiment properly because it processes contextual information together with word relations. The proposed model the BiLSTM architecture features its optimization capabilities for feature extraction while speeding up training time to reach accurate results. The model delivers exceptional

| Model | F1-Score | MSE |
|---|---|---|
| Classifier Neural Network + Multilingual BERT (Furniturewala et al., 2022) | 0.234 | 3.000 |
| Bi-LSTM (Guha et al., 2022) | 0.098 | 6.000 |
| M-BERT (Srivastava and Singh, 2021b) | 0.202 | 2.797 |
| **Proposed BiLSTM with SMOTE** | **0.742** | **0.106** |

Table 1: Quantitative Analysis of Dataset 1

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score |
|---|---|---|---|---|
| HF-CSA (Raza et al., 2023) | 76.18 | 81.61 | 73.86 | 0.234 |
| CNN (Angel et al., 2020) | | 51.00 | 49.60 | 0.458 |
| GRU (Angel et al., 2020) | | 35.80 | 37.30 | 0.290 |
| **Proposed BiLSTM with SMOTE** | **42.41** | **48.78** | **11.10** | **0.390** |

Table 2: Quantitative Analysis of Dataset 2

| Model | Accuracy (%) | Precision (%) | Recall (%) | F1-Score |
|---|---|---|---|---|
| SVC (Ahmed and Ahmed, 2024) | 91.71 | 91.81 | 91.76 | 0.916 |
| KN (Ahmed and Ahmed, 2024) | 72.05 | 75.13 | 71.93 | 0.715 |
| NB (Ahmed and Ahmed, 2024) | 79.14 | 79.18 | 79.19 | 0.791 |
| DT (Ahmed and Ahmed, 2024) | 87.14 | 87.80 | 87.25 | 0.868 |
| LR (Ahmed and Ahmed, 2024) | 88.62 | 88.74 | 88.64 | 0.885 |
| RF (Ahmed and Ahmed, 2024) | 92.13 | 92.15 | 92.14 | 0.920 |
| AdaBoost (Ahmed and Ahmed, 2024) | 67.90 | 68.41 | 67.78 | 0.676 |
| BgC (Ahmed and Ahmed, 2024) | 88.41 | 88.85 | 88.51 | 0.881 |
| ETC (Ahmed and Ahmed, 2024) | 92.97 | 93.06 | 92.92 | 0.929 |
| GBDT (Ahmed and Ahmed, 2024) | 72.40 | 73.32 | 72.28 | 0.721 |
| XGB (Ahmed and Ahmed, 2024) | 86.58 | 86.92 | 86.59 | 0.864 |
| **Proposed BiLSTM with SMOTE** | **87.29** | **87.73** | **86.86** | **0.870** |

Table 3: Quantitative Analysis of Dataset 3

performance with accuracy of 74.24%, precision of 75.85% with recall value of 73.42% , F1-Score of 0.742 & MSE of 0.106 for dataset 1. For dataset 2, the values are 42.41%, 48.78%, 11.10%, 0.390 & 0.215 respectively. If we look at dataset 3, the values vary from 87.29%, 87.73%, 86.86%, 0.870 & 0.215 respectively.

Future studies should focus their research on enhancing model generalization and robustness through specific improvement areas which these current constraints have identified. The model needs domain adaptation strategies or transfer learning approaches to achieve better generalization between different Hinglish usage patterns.

## References

Sabbir Ahmed and Md Sazzadur Ahmed. 2024. Hinglish products review-based sentiment analysis on using ml algorithms. In *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS)*, pages 1–7. IEEE.

Manar Alfreihat, Omar Almousa, Yahya Tashtoush, Anas AlSobeh, Khalid Mansour, and Hazem Migdady. 2024. Emo-sl framework: emoji sentiment lexicon using text-based features and machine learning for sentiment analysis. *IEEE Access*.

Ohud Alsemaree, Atm S Alam, Sukhpal Singh Gill, and Steve Uhlig. 2024. Sentiment analysis of arabic social media texts: A machine learning approach to deciphering customer perceptions. *Heliyon*, 10(9).

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2023. Improving multilingual neural machine translation system for indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(6):1–24.

Azhar Ahmed Bilal, O Ayhan Erdem, and Sinan Toklu. 2024. Children's sentiment analysis from texts by using weight updated tuned with random forest classification. *IEEE Access*.

Sudhansu Bala Das, Samujjal Choudhury, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025. Investigating the effect of backtranslation for indic languages. In *Proceedings of the First Workshop on Natural Language*

*Processing for Indo-Aryan and Dravidian Languages*, pages 152–165.

Shaz Furniturewala, Vijay Kumari, Amulya Ratna Dash, Hriday Kedia, and Yashvardhan Sharma. 2022. Bits pilani at hinglisheval: Quality evaluation for code-mixed hinglish text using transformers. *arXiv preprint arXiv:2206.08680*.

Prantik Guha, Rudra Dhar, and Dipankar Das. 2022. Ju_nlp at hinglisheval: Quality evaluation of the low-resource code-mixed hinglish text. *arXiv preprint arXiv:2206.08053*.

Aixiang He and Mideth Abisado. 2024. Text sentiment analysis of douban film short comments based on bert-cnn-bilstm-att model. *Ieee Access*.

Zhengbing Hu, Ivan Dychka, Kateryna Potapova, and Vasyl Meliukh. 2024. Augmenting sentiment analysis prediction in binary text classification through advanced natural language processing models and classifiers. *Int. J. Inf. Technol. Comput. Sci*, 16:16–31.

Adarsh Singh Jadon, Mahesh Parmar, and Rohit Agrawal. 2024. Hinglish sentiment analysis: Deep learning models for nuanced sentiment classification in multilingual digital communication. In *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, pages 318–323. IEEE.

Abhidha Joshi, Rohit Agrawal, Nitin Arvind Shelke, and Adarsh Singh Jadon. 2025. Effective content moderation: A framework for toxic comment analysis using machine learning. In *2025 International Conference on Next Generation Information System Engineering (NGISE)*, volume 1, pages 1–7.

Hongchan Li, Yantong Lu, and Haodong Zhu. 2024. Multi-modal sentiment analysis based on image and text fusion based on cross-attention mechanism. *Electronics*, 13(11):2069.

Sihong Li and Jinglong Chen. 2024. Virtual human on social media: Text mining and sentiment analysis. *Technology in Society*, 78:102666.

Zhizhong Liu, Bin Zhou, Dianhui Chu, Yuhang Sun, and Lingqiang Meng. 2024. Modality translation-based multimodal sentiment analysis under uncertain missing modalities. *Information Fusion*, 101:101973.

Juan Antonio Lossio-Ventura, Rachel Weger, Angela Y Lee, Emily P Guinee, Joyce Chung, Lauren Atlas, Eleni Linos, and Francisco Pereira. 2024. A comparison of chatgpt and fine-tuned open pre-trained transformers (opt) against widely used sentiment analysis tools: sentiment analysis of covid-19 survey data. *JMIR Mental Health*, 11:e50150.

Hui Qi Low, Pantea Keikhosrokiani, and Moussa Pourya Asl. 2024. Decoding violence against women: analysing harassment in middle eastern literature with machine learning and sentiment analysis. *Humanities and Social Sciences Communications*, 11(1):1–18.

Tanjim Mahmud, Rezaul Karim, Rishita Chakma, Tanjia Chowdhury, Mohammad Shahadat Hossain, and Karl Andersson. 2024. A benchmark dataset for cricket sentiment analysis in bangla social media text. *Procedia Computer Science*, 238:377–384.

Poonam Narang, Ajay Vikram Singh, and Himanshu Monga. 2024. Enhanced detection of fabricated news through sentiment analysis and text feature extraction. *International Journal of Information Technology*, 16(6):3891–3900.

Mina Ramzy and Bahaa Ibrahim. 2024. User satisfaction with arabic covid-19 apps: Sentiment analysis of users' reviews using machine learning techniques. *Information Processing & Management*, 61(3):103644.

Vivek Srivastava and Mayank Singh. 2021a. Hinge: A dataset for generation and evaluation of code-mixed hinglish text. *arXiv preprint arXiv:2107.03760*.

Vivek Srivastava and Mayank Singh. 2021b. Quality evaluation of the low-resource synthetically generated code-mixed hinglish text. *arXiv preprint arXiv:2108.01861*.

Shuo Yin and Guoqiang Zhong. 2024. Textgt: A double-view graph transformer on text for aspect-based sentiment analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 19404–19412.

# Automatic Animacy Classification for Latvian Nouns

**Ralfs Brutāns**
University of Amsterdam
Vrije Universiteit Amsterdam
ralfs.brutans@student.uva.nl

**Jelke Bloem**
University of Amsterdam
j.bloem@uva.nl

## Abstract

We introduce the first automatic animacy classifier for the Latvian language. Animacy, a linguistic feature indicating whether a noun refers to a living entity, plays an important role in Latvian grammatical structures and syntactic agreement, but remains unexplored in Latvian NLP. We adapt and extend existing methods to develop type-based animacy classifiers that distinguish between human and non-human nouns. Due to the limited utility of Latvian WordNet, the classifier's training data was derived from the WordNets of Lithuanian, English, and Japanese. These lists were intersected and mapped to Latvian nouns from the Tēzaurs dictionary through automatic translation. The resulting dataset was used to train classifiers with fastText and LVBERT embeddings. Results show good performance from a MLP classifier using the last four layers of LVBERT, with Lithuanian data contributing more than English. This demonstrates a viable method for animacy classification in languages lacking robust lexical resources and shows potential for broader application in morphologically rich, under-resourced languages.

## 1 Introduction

There are many languages that are bound to go extinct despite efforts to preserve them, while others, such as English, are widely spoken and face no such risk. The Latvian language is positioned somewhere in between (Jansone, 2010), meaning that its long-term survival depends on active efforts to maintain and develop it. As the official language of Latvia and one of the official languages of the European Union, Latvian has around 1.5 million native speakers[1], significantly fewer than global languages like English. This smaller speaker base also means that Latvian is considerably less researched in fields such as natural language processing (NLP) (Laucis and Jēkabsons, 2021). Ensuring that Latvian keeps pace with advances in NLP is essential not only for preserving and modernizing the language but also for supporting its use in digital applications such as machine translation and automated text processing.

One important but underexplored linguistic feature in NLP research, particularly for morphologically rich languages like Latvian, is animacy. Animacy refers to how "alive" or independently acting a noun's referent is — humans and animals are animate, while objects are not. This distinction is encoded in the grammars of many natural languages, influencing word order, case marking, and agreement patterns. Studies suggest that incorporating animacy into computational models can enhance machine translation and parsing accuracy (Øvrelid, 2006, 2008), in addition to informing linguistic studies. However, for Latvian, we are not aware of the existence of any animacy classifier.

We present an approach to animacy classification for Latvian nouns based on static and contextual word embeddings. While other animacy classifiers for under-resourced languages have used this approach (Tepei and Bloem, 2024 for Romanian), our approach is novel in relying on lexical-semantic resources for higher-resource languages. Previous approaches rely on WordNet hypernym relations to obtain a seed set of animacy-labeled data for supervised learning, but as the Latvian WordNet does not yet have a full tree of hypernym and hyponym relations, we instead rely on the WordNets for higher-resource languages and automatic translation to obtain such data.

## 2 Related Work

Three types of animacy are usually distinguished: grammatical, biological and conceptual (de Swart

[1] https://valoda.lv/valsts-valoda/



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 90–97, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-011

and de Hoop, 2018). Entities that possess physical characteristics such as the ability to die are said to as biologically animate. The speaker's perspective and cultural upbringing serve as the foundation for conceptual animacy. The way that people personify or give non-living things agency reflects this, for example, in mythology. Grammatical animacy, however, illustrates how a language's grammar reflects both biological and conceptual animacy. It functions as a condition or semantic feature that affects linguistic structures such as case marking and verb agreement. Usually, the concept of animacy is seen as being a continuous scale ranging from humans to inanimate abstract objects. According to Yamamoto (2006, p. 36), animacy is a matter of gradience determined by the overall animacy scale, hierarchy of persons, the agency scale, and the individuation scale.

Despite animacy usually being viewed as existing on a continuous scale or a hierarchy (DeLancey, 1981), natural language requires that concepts be categorized. The two most simple categorizations would be distinguishing between animate and inanimate nouns or distinguishing between human and non-human nouns. Most effects of grammatical animacy are based on a binary split, with tripartite systems being rare. As a consequence, most NLP literature on animacy also discusses the distinction between two to three categories.

Latvian does not mark animacy distinctions directly, but animacy has several effects on the grammaticality and felicitousness of sentences. Latvian has a somewhat free word order; however, there are more and less common sentence structures. Even though all six word order variations for subject-verb-object placement are possible, the most common structure is subject-verb-object (SVO) when the subject is animate, and object-verb-subject (OVS) is more frequent with inanimate subjects (Voits, 2014). Seržant and Taperte (2016) found that animacy plays a role in influencing the choice between accusative and nominative case marking in the Latvian debitive construction. Animate NPs also appear to be more likely to trigger genitive agreement, where the predicate agrees with the noun in the genitive case rather than with the quantifier. Inanimate NPs, on the other hand, do not favor the genitive or quantifier agreement (Kalnača and Lokmane, 2022, p. 85). These interactions between animacy and the felicitousness and grammaticality of sentences in Latvian suggest that animacy could be a useful feature also for downstream NLP tasks for Latvian, such as coreference resolution.

Animacy classifiers have been made for other languages, including under-resourced ones. Some of the first research on automatic animacy classification for nouns was done by Øvrelid (2004) on animacy classifiers for Norwegian (Øvrelid, 2005) and for Swedish (Øvrelid, 2008; Øvrelid, 2009). These classifiers are based on morphosyntactic features that were selected on linguistic grounds to classify into binary animate/inanimate categories. These classifiers achieved quite good results, achieving up to a 98.6% accuracy on unseen nouns. However, they are based on large pre-annotated animacy corpora, which is not something available for many under-resourced languages.

Subsequently, Bowman and Chopra (2012) proposed a classifier that classifies nouns into ten categories. This paper highlights the problem of trying to classify nouns into more categories than are expressed in grammar. It is more difficult to discriminate animacy when grammaticality and felicitousness is only governed by human/non-human or animate/inanimate categories as models can then only rely on semantic cues, not syntactic or morphological ones. Bloem and Bouma (2013) present an animacy classification tool for Dutch, which combines type-based classification using distributional features with a seed set of noun types that were given an animacy label based on the Cornetto lexical-semantic database (Vossen, 2006). They tested classification with a two-way distinction (human/nonhuman) and a three-way distinction (animate human/animate nonhuman/inanimate). Classification for the human/nonhuman distinction, which corresponds to distinctions made in Dutch grammar, performed much better. Their best-performing classification algorithm was a K-nearest neighbour classifier.

More recent approaches have turned to transfer learning to overcome data scarcity and enhance generalization across tasks and languages. Transfer learning has been widely used in natural language processing to address the challenges posed by limited labeled data, especially in under-resourced languages. The approach involves reusing representations learned from a general task, such as language modeling, for more specific tasks like animacy classification. Pretrained models such as FastText (Bojanowski et al., 2017) and contextual models like BERT (Devlin et al., 2019) are commonly used. For Latvian, LVBERT (Znotiņš and Barzdiņš, 2020) provides pretrained embeddings that can be applied to downstream tasks with minimal task-specific training. Prior work suggests that transfer learning can support tasks requiring semantic generalization by utilizing knowledge encoded during pretraining (Ruder et al., 2019; Conneau et al., 2020). In the case of animacy, this includes properties such as agency and sentience, which may be implicitly captured by language models.

## 3 Methodology

### 3.1 Use of WordNets for animacy-annotated lists

Based on the results of previous work and because Latvian does not have a strict grammatical animacy distinction, the animacy classifier we make distinguishes between human and non-human classes, employing lemmas from a word list to make a type-based classifier. Inspired by the recent animacy classification work on the under-resourced Romanian by Tepei and Bloem (2024), we used hyponymy relations to extract a seed set of nouns that were labeled for animacy using WordNet.

This labeling stems from the hierarchical structure of WordNet: It consists of a hierarchy of synsets that are in hyponym and hypernym relations with each other, with specific synsets at the bottom and general synsets at the top, representing concepts such as *entity* and *event*. Under *entity*, we find concepts such as *life form* and *object*. By taking words from all synsets that are hyponyms of one such high-level synset, such as the one corresponding to human words, it is possible to obtain a large number of words that refer to human entities.

However, Latvian WordNet (Paikens et al., 2023) is still under construction and is largely lacking in hyponymy and hypernymy relations, not containing a full tree of synsets. Therefore, we developed a different approach based on higher-resource WordNets. More specifically, WordNets for three languages— Lithuanian (Garabík and Pileckyte, 2013), English (Fellbaum, 1998) and Japanese (Bond and Kuribayashi, 2023) were used to extract lists of human and non-human nouns. The English WordNet was used for its interpretability, and Japanese WordNet was used for its large word tree containing a vast amount of noun relations and for the fact that it is not an Indo-European language, typologically distinct from both Latvian and English. In contrast, the Lithuanian WordNet was used because of its similarity with Latvian, the only other living Baltic language.

As in previous work, we constructed the seed sets of words with animacy labels by identifying high-order hypernyms that contained no instance of the other class for the three languages, namely, *asmuo, person* and 人 (all with the meaning person) as the human targets for Lithuanian, English, and Japanese, respectively. For the non-human class, all other unique beginner synsets that do not contain person as a hyponym were used for the inanimate class. After having established these high-order hypernym synsets, lists of all their hyponyms were extracted to obtain human and non-human nouns for all three languages.

### 3.2 Translating Latvian nouns

We used the online lexical resource Tēzaurs (Spektors et al., 2025) to obtain a dataset of Latvian nouns. All of these unique noun lemmas were automatically translated to Lithuanian, English, and Japanese for comparison with the extracted lists of animacy-labeled lists of nouns. To translate these nouns, we used the Google Translate API[2], which was found by Rikters (2015) to perform well for English-Latvian translation at the time.

The translations of the nouns were checked against the animacy-labeled lists of nouns in the three languages, and if a word was present in all three animacy-labeled sets with the same label, then its Latvian counterpart was included in an animacy-labeled list for Latvian with the corresponding label of human or non-human. This restriction reduces the possibility that translation errors between particular language pairs affect the quality of our seed set, as there was no manual translation quality control and polysemous words could have been translated incorrectly. In the end, the list consists of 5183 nouns, of which 735 are labeled as human.

### 3.3 FastText-based classifiers

We used pre-trained fastText embeddings (Bojanowski et al., 2017) to obtain static word vectors for Latvian nouns. FastText was chosen for its subword modeling capabilities, making it effective for morphologically rich languages, and prior work found it to be the best-performing static embedding for Latvian (Laucis and Jēkabsons, 2021).

[2]The API was called using the deep-translator library for Python: `https://pypi.org/project/deep-translator/`

Vectors were 300-dimensional, with character n-grams of length 5, a window size of 5, and 10 negative samples, trained on Common Crawl and Wikipedia. Each vector was paired with the animacy label derived from the WordNet intersection. We trained classifiers using K-nearest neighbors (KNN) ($k = 5$), Random Forest (RF) (100 estimators, Gini criterion), and Multi-Layer Perceptron (MLP) (hidden size 100, $\alpha = 0.0001$, learning rate 0.001) algorithms. We used these classifiers because they were used by Tepei and Bloem (2024), who included them based on good performance in previous work.

### 3.4 LVBERT-based classifiers

To explore the potential of contextual embeddings, we used LVBERT (Znotiņš and Barzdiņš, 2020), a transformer model trained on Latvian corpora. For each noun token, we extracted either layer 0 (non-contextualized) or a 3072-dimensional vector from the concatenation of the final four layers (layers 9-12, following Hosseini et al. 2023).

We try the latter approach because deeper layers capture richer semantic information (Devlin et al., 2019), and concatenation has shown strong performance in semantic similarity tasks. The first non-special token (i.e., the noun) was used for classification. Layer 0 embeddings were also tested to compare performance with fastText, as lower transformer layers may behave like static embeddings (Vulić et al., 2021). We cannot tune LVBERT for the task with a token classifier head as no corpus with animacy annotation in context is available.

### 3.5 Evaluation methods

To assess the quality of the static and contextual embedding-based classifiers, we split the labeled noun types into an 80%/20% train/test set, which was then used to perform a type-based evaluation of the classifiers.

We also perform a token-based evaluation because it represents a more naturalistic use setting despite the classifier being type-based. To this end, we chose, compiled, and cleaned nine random Wikipedia articles. Using the Python library Stanza (Qi et al., 2020), which has a POS tagger for Latvian trained on the Universal Dependencies treebanks for Latvian (Pretkalniņa et al., 2018), a list of nouns present in the texts was extracted and manually annotated for animacy by a native Latvian speaker. As the classifier is type-based, the nouns are lemmatized before annotation and prediction. Lemmas representing human collectives (e.g., *valdība* 'government') were assigned the non-human category due to them being treated as inanimate in Latvian grammar, exemplified by the use of demonstrative pronouns instead of personal pronouns. The classifiers were then used to predict class membership for the given nouns. Although the classifier is type-based and does not consider the surrounding context of nouns, token-based evaluation can provide a better benchmark of the classifier's performance in a naturalistic setting.

## 4 Results

### 4.1 Type-based evaluation

#### 4.1.1 Results for the classifiers made with fastText embeddings

| Classifier | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| KNN | 0.857 | 0.780 | 0.222 | 0.345 |
| RF | 0.878 | **0.915** | 0.307 | 0.461 |
| MLP | **0.900** | 0.728 | **0.653** | **0.688** |

Table 1: Type-based evaluation performance of fastText-based classifiers. Baseline accuracy is 0.830.

For fastText-based classification of noun types, the RF algorithm achieves a higher precision score of 91.5% against the KNN and MLP models (see table 1). However, the MLP algorithm shows better recall and accuracy scores of 65.3% and 90.0%, respectively. This entails that when the RF model predicts the human class, it is almost always correct (precision); however, it is very conservative in labeling nouns as human, leading to very low recall of 30.7% (false negatives). On the other hand, the high accuracy and recall scores for MLP show that it is overall quite good at predicting class membership, and it achieves the highest F1 score of 68.8%. KNN shows the worst performance overall. Baseline accuracy for this dataset is 83%, as 83% of the test nouns are non-human.

#### 4.1.2 Results for classifiers trained with LVBERT embeddings

Using layer 0 embeddings from LVBERT (Znotiņš and Barzdiņš, 2020) for classifier training did not prove to be useful (see table 2), yielding worse scores than their fastText counterparts (with the exception of KNN).

Results with the last four layers of LVBERT embeddings are better, with the MLP classifier clearly outperforming the KNN and RF algorithms.

| Classifier | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| KNN | 0.854 | 0.602 | 0.420 | 0.494 |
| RF | 0.842 | 0.620 | 0.176 | 0.274 |
| MLP | **0.880** | **0.697** | **0.523** | **0.597** |

Table 2: Type-based evaluation on LVBERT layer 0-based classifiers. Baseline accuracy is 0.830.

| Classifier | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| KNN | 0.905 | 0.824 | 0.557 | 0.664 |
| RF | 0.849 | **0.913** | 0.119 | 0.210 |
| MLP | **0.916** | 0.795 | **0.682** | **0.734** |

Table 3: Type-based evaluation performance of LVBERT last four layer concatenation-based classifiers

| Classifier | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| KNN | 0.905 | 0.717 | 0.349 | 0.469 |
| RF | **0.931** | **0.911** | 0.468 | 0.618 |
| MLP | 0.894 | 0.542 | **0.771** | **0.636** |

Table 4: Token-based evaluation on fastText-based classifiers. Baseline accuracy is 0.880.

| Classifier | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| KNN | 0.910 | 0.636 | 0.578 | 0.606 |
| RF | 0.896 | **1.000** | 0.138 | 0.242 |
| MLP | **0.938** | 0.768 | **0.697** | **0.731** |

Table 5: Token-based evaluation on LVBERT-based classifiers

As shown in Table 3, the MLP classifier achieves the highest accuracy (0.916), recall (0.682), and F1 score (0.734) among the three LVBERT-based classifiers, along with a strong precision score (0.795). This indicates that the LVBERT-based MLP classifier is both relatively accurate and balanced in predicting the "human" class. The RF classifier, while achieving the highest precision (0.913) among the three, performs poorly in recall (0.119) and F1 score (0.210). This reflects a conservative approach in labeling tokens as "human," resulting in many false negatives. The KNN classifier surpasses the fastText-based KNN model. The LVBERT-based MLP classifier outperforms the fastText-based MLP classifier. Only the LVBERT-based RF classifier does not outperform its fastText-based counterpart. The performance dynamics of the different algorithms remain the same in a type-based evaluation, where the RF algorithm has the highest precision, while MLP outperforms on the other three metrics. This suggests that richer, contextualized representations from transformer models are beneficial when classifying noun animacy at the type level.

### 4.2 Token-based evaluation

For the token-based evaluation with LVBERT, we only used the last four layer approach due to superior performance in the type-based evaluation. This evaluation aims to show whether a more naturalistic setting would affect the performance rankings of the classifiers. Nine random Wikipedia articles on different topics were chosen preprocessed. Next, a Latvian POS tagger trained on the UD (universal dependencies) treebank corpus for Latvian (Pretkalniņa et al., 2018) was employed using the Stanza (Qi et al., 2020) library for Python to obtain a list of 1342 noun lemmas for animacy labeling. All the lemmas were manually annotated by a native Latvian speaker with human/non-human labels based on the meaning of the word token in context. 46 lemmas were excluded from this test set due to POS-tagging errors or faulty text to obtain 1296 annotated lemmas, of which 908 were used for unseen prediction. Out of these 908 lemmas, 799 were annotated with the non-human label and 108 with the human label, setting the majority baseline accuracy at 88.0%

In this evaluation, the LVBERT-based classifiers generally outperform the fastText-based classifiers. The LVBERT-based MLP classifier achieved the highest accuracy (0.938) and F1 score (0.731) across all settings. It also had the highest recall (0.697), indicating stronger performance in identifying human-referent nouns. The LVBERT-based RF classifier, while achieving perfect precision (1.000), showed a very low recall (0.138), suggesting a highly conservative classification strategy that avoids false positives but misses many actual human nouns. For fastText, overall results are lower but still competitive, and RF classifiers perform better here than they do on LVBERT embeddings.

### 4.3 Language resource ablation

As our methodology involves combining data from higher-resource WordNets, we also evaluate the contribution of each source language WordNet by training and testing classifiers using only one of the languages as source data. We perform the token-based evaluation for all nouns that are not in the training data (unseen nouns). This does mean that each classifier has a different test set, as some lan-

| Language | Acc. | Pre. | Rec. | F1 |
|---|---|---|---|---|
| Lithuanian | **0.932** | **0.680** | **0.742** | **0.710** |
| English | 0.862 | 0.383 | 0.561 | 0.455 |
| Japanese | 0.908 | 0.632 | 0.655 | 0.643 |

Table 6: Token-based evaluation on LVBERT-based MLP classifiers trained on single language WordNets. Baseline accuracy differs per language.

guages have labels for more nouns than others, so the results are not directly comparable, but it does give an impression of the relative contribution of each resource. Specifically, for the English-based classifier there are 398 unseen nouns in our evaluation set, for Japanese there are 434 and for Lithuanian (the smallest WordNet) there are 790. For comparison, the original token-based evaluation had 908 unseen nouns (not occurring in all three resources). We perform this experiment in the best-performing setting, using the last four layers of LVBERT with the MLP classifier. The results are shown in Table 6. We observe that the classifier based on Lithuanian WordNet outperforms the others, despite this resource being the smallest (6357 noun synsets, compared to 82,115 for English and 42,737 for Japanese). Latvian and Lithuanian are closely related typologically, with both being East Baltic languages. This result suggests that typological relatedness is more beneficial than resource comprehensiveness for transfer learning for animacy classification in a natural language setting. However, the approach of combining all three language resources still outperforms Lithuanian only (0.731 vs 0.710 F1 score).

## 5 Discussion

This study introduced the first classifiers for predicting animacy (human vs. non-human) in Latvian nouns, using a methodology adapted for a low-resource setting. We evaluated 12 classifiers based on fastText and LVBERT embeddings, with animacy-labeled training data derived through multilingual WordNet intersection and translation. While we found that training data from a typologically related language was more useful, the best results were achieved by LVBERT-based MLP classifiers using the final four layers of the model trained on labels from an intersection of three languages' WordNets. These outperformed fastText-based models in both type- and token-based evaluations, with the best model achieving 93.8% accuracy on unseen nouns.

Although all classifiers were trained on type-level data, token-based evaluation showed that contextual embeddings can generalize well to more naturalistic usage, even without explicit token-level supervision. Layer 0 LVBERT embeddings, which behave more like static vectors, underperformed compared to deeper contextual layers. The success of LVBERT shows that transformer-based representations can be beneficial even in the absence of large annotated corpora.

Another promising direction is to use generative large language models' zero-shot generalization capability. Recent work demonstrates that GPT-3 can distinguish animate/inanimate entities in zero-shot settings across languages (Pucci et al., 2025), though this has not yet been explored in a classification task. Probing or fine-tuning LLMs such as LVBERT, LitLat BERT, or multilingual open-weight models (e.g., Gemma, LLaMA) on animacy tasks could offer new insights and performance improvements. Evaluating how well such models generalize animacy features to under-resourced languages would help clarify their linguistic competence and applicability in downstream NLP tasks.

## 6 Conclusion

We present the first type-based approach to animacy classification for Latvian nouns using cross-lingual projection and multilingual lexical resources. Animacy-labeled word lists were automatically constructed by aligning English, Lithuanian, and Japanese WordNets with Latvian nouns from the Tēzaurs dictionary via translation. This enabled training data creation without manual annotation. We trained classifiers using fastText and contextual LVBERT embeddings. Results showed that LVBERT-based models—especially MLP with concatenated final layers—outperformed fastText models in both type- and token-based evaluations. While RF classifiers achieved the highest precision, MLPs offered better balance overall. A language ablation study showed the most typologically related language to contribute more.

This work demonstrates the feasibility of animacy classification in low-resource languages without native WordNets. Despite limitations—such as label noise from translation and lack of context in static embeddings—our approach lays a foundation for extending animacy annotation and classification to other languages.

## 7 Limitations

Several limitations remain. Training data labels were derived via automatic translation, which may introduce noise. Furthermore, type-based classifiers cannot resolve context-sensitive cases of animacy, such as polysemous words (*medijs*: psychic or media in Latvian). In a naturalistic setting, our classifier would have to be used after lemmatization, and imperfect lemmatization due to the extensively inflected nature of Latvian might reduce accuracy. Future work could focus on building token-level classifiers, such as by tuning LVBERT. This would require a corpus where nouns are annotated for animacy in context, which is currently unavailable for Latvian. Another direction would be to address the class imbalance in training data by augmenting the human noun class through synonym expansion.

## 8 Ethical considerations

We do not foresee any particular harmful impacts of this work. While the pre-trained embeddings we use may encode harmful biases, we could not identify any reason to assume that these biases pertain to the human/nonhuman distinction that we classify. Most concerns regarding bias identified in the literature pertain to social identities that differ between humans (e.g. gender bias). When deploying animacy classification of the type we propose, we do recommend to evaluate that people with protected characteristics relevant to the use case aren't more likely to be misclassified as nonhuman, as this may cause harm.

## References


Jelke Bloem and Gosse Bouma. 2013. Automatic animacy classification for Dutch. 3:82–102.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *arXiv:1607.04606 [cs]*.

Francis Bond and Takayuki Kuribayashi. 2023. The Japanese WordNet 2.0. *ACL Anthology*, pages 179–186.

Samuel R. Bowman and Harshit Chopra. 2012. Automatic Animacy classification. In *Proceedings of the NAACL HLT 2012 Student Research Workshop*, pages 7–10, Montréal, Canada. Association for Computational Linguistics.

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.

Scott DeLancey. 1981. An interpretation of split ergativity and related patterns. *Language*, 57:626–657.

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, pages 4171–4186.

Christiane Fellbaum. 1998. *WordNet : an electronic lexical database*. Mit Press.

Radovan Garabík and Indre Pileckyte. 2013. From multilingual dictionary to Lithuanian WordNet.

MohammadSaleh Hosseini, Munawara Munia, and Latifur Khan. 2023. BERT has more to offer: BERT layers combination yields better sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15419–15431, Singapore. Association for Computational Linguistics.

Ilga Jansone. 2010. Latvian. *Revue belge de philologie et d'histoire*, 88(3):741–764.

Andra Kalnača and Ilze Lokmane. 2022. Partitive genitive constructions and agreement variations in Latvian. *Linguistic Variation*.

Rolands Laucis and Gints Jēkabsons. 2021. Evaluation of word embedding models in Latvian NLP tasks based on publicly available corpora. *Applied Computer Systems*, 26:132–138.

Lilja Øvrelid. 2005. Animacy classification based on morphosyntactic corpus frequencies: some experiments with Norwegian nouns. In *Proceedings of the Workshop on Exploring Syntactically Annotated Corpora*, pages 24–34.

Lilja Øvrelid. 2009. Empirical evaluations of Animacy annotation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 630–638, Athens, Greece. Association for Computational Linguistics.

Peteris Paikens, Agute Klints, Ilze Lokmane, Lauma Pretkalniņa, Laura Rituma, Madara Stāde, and Laine Strankale. 2023. Latvian WordNet. *ACL Anthology*, pages 187–196.

Lauma Pretkalniņa, Laura Rituma, and Baiba Saulīte. 2018. Deriving enhanced universal dependencies from a hybrid dependency-constituency treebank. *Lecture notes in computer science*, pages 95–105.

Giulia Pucci, Fabio Massimo Zanzotto, and Leonardo Ranaldi. 2025. Animate, or inanimate, that is the question for large language models. *Information*, 16(6):493.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.

Matīss Rikters. 2015. Multi-system machine translation using online apis for English-Latvian.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilja A Seržant and Jana Taperte. 2016. Differential argument marking with the Latvian debitive. pages 199–258.

Andrejs Spektors, Lauma Pretkalniņa, Normunds Grūzītis, Pēteris Paikens, Laura Rituma, Baiba Saulīte, Gunta Nešpore-Bērzkalne, Ilze Lokmane, Agute Klints, Madara Stāde, Mikus Grasmanis, Ilze Auziņa, Artūrs Znotiņš, Roberts Darģis, and Guntis Bārzdiņš. 2025. Tēzaurs.lv 2025 (spring edition). *Clarin.lv*.

Peter de Swart and Helen de Hoop. 2018. Shifting animacy. *Theoretical Linguistics*, 44:1–23.

Maria Tepei and Jelke Bloem. 2024. Automatic animacy classification for Romanian nouns. *ACL Anthology*, pages 1825–1831.

Toms Voits. 2014. Discourse-related word order variation in Latvian. *Valoda: nozīme un forma*, pages 144–156.

PTJM Vossen. 2006. Cornetto: Een lexicaal-semantische database voor taaltechnologie. *Dixit*, (special issue).

Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2021. LexFit: Lexical fine-tuning of pretrained language models. 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 5269–5283, Online. Association for Computational Linguistics.

Mutsumi Yamamoto. 2006. *Agency and impersonality : their linguistic and cultural manifestations*. J. Benjamins Pub. Co.

Artūrs Znotiņš and Guntis Barzdiņš. 2020. LVBERT: Transformer-based model for Latvian language understanding. *Frontiers in artificial intelligence and applications*.

Lilja Øvrelid. 2004. Disambiguation of syntactic functions in norwegian: Modeling variation in word order interpretations conditioned by animacy and definiteness. *Proceedings of the 20th Scandinavian Conference of Linguistics*.

Lilja Øvrelid. 2006. Towards robust animacy classification using morphosyntactic distributional features. *ACL Anthology*, Student Research Workshop:47–54.

Lilja Øvrelid. 2008. Linguistic features in data-driven dependency parsing. *ACL Anthology*, pages 25–32.

# Bootstrapping a Sentence-Level Corpus Quality Classifier for Web Text using Active Learning

Maximilian Bley[1], Thomas Eckart[2], and Christopher Schröder[1,3]

[1]Institute for Applied Informatics (InfAI) at Leipzig University
[2]Saxon Academy of Sciences and Humanities, Leipzig
[3]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig

## Abstract

The quality of training data is an essential factor for training large language models (LLMs) as it directly impacts their performance. While high-quality data is crucial for training competitive LLMs, existing preprocessing pipelines still partly rely on rules, which are computationally cheap but also inherently limited to simpler patterns. Model-based filtering on the other hand, is more flexible and can detect finer-grained patterns and semantics, but often requires substantial amounts of labeled data. While there are models for common problems (such as toxicity classification), this is often only the case for resource-rich languages and well-studied problems—leaving gaps in coverage for other languages, problems, or combinations thereof. In this work, we investigate the feasibility of model-based preprocessing despite the absence of labeled data. We use active learning to bootstrap a sentence-level multi-label classifier that detects textual problems of traditional text cleaning approaches. With only 498 examples, the final classifier reaches macro- and micro-$F_1$ scores of 0.80 and 0.84, making it suitable for practical use. Moreover, we find that it captured subtle errors compared to a rule-based baseline. We publish the training code, a labeled corpus quality classification dataset, and the resulting classifier[1].

## 1 Introduction

Pre-training large language models (LLMs) requires not only vast amounts of textual data but also high-quality content, as recent studies show the impact of data quality on downstream performance (Raffel et al., 2020; Penedo et al., 2023; Longpre et al., 2024; Li et al., 2024).

While there have been many efforts to curate and clean LLM pre-training corpora, only some of the possible steps use model-based approaches such as language identification (Joulin et al., 2016; Grave et al., 2018), perplexity-based filtering (Ankner et al., 2025; Thrush et al., 2025), predicting similarity to reference text (Li et al., 2024), toxic or adult content detection (Soldaini et al., 2024), or the targeted search for certain contents such as educational texts (Wettig et al., 2024). However, there still are various preprocessing steps that resemble traditional text cleaning. They target noise that usually results from text extraction artifacts in web corpora such as, among others, incorrectly formatted text, non-linguistic content, random word sequences, letter spacing, encoding errors, or repeating characters, and are still predominantly rule-based (Albalak et al., 2024). Developing such rules is known to be time-consuming, often highly tailored to specific domains and languages, and may fail to capture more subtle issues compared to supervised models (Laurençon et al., 2022; Longpre et al., 2024; Henriksson et al., 2025).

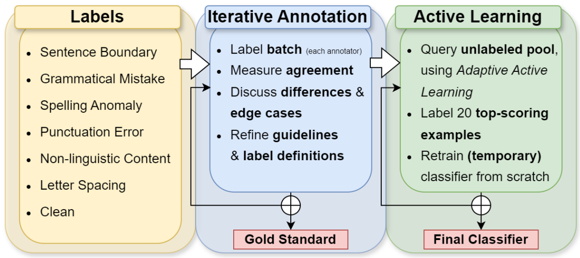



Figure 1: The development process of our approach. We begin by defining seven corpus quality labels along with their annotation guidelines, then we annotate a gold standard for evaluation, and finally train a corpus quality classifier with active learning.

Supervised learning, however, requires a substantial amount of training data. While there are some existing datasets for tasks such as toxic or adult content detection, they only cover a limited number of languages, and moreover, to the best of our knowledge, none of them address the problems that are usually handled by rule-based filtering. With

[1]https://github.com/maximilian-bley/german-webtext-quality-classification



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 98–109, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-012

smaller[2] language models[3] becoming increasingly effective, we argue that it is time to widen the scope of model-based preprocessing.

To investigate the feasibility of model-based preprocessing, we train a supervised corpus quality classifier with seven classes: six representing distinct types of textual deficiencies and one capturing the absence of those.[4] Since no training data exists for this particular task, we apply *active learning*, an iterative approach that aims to minimize annotation effort. We begin by developing a classification scheme and corresponding annotation guidelines. To evaluate the resulting classifier, we create a gold standard through iterative annotation, optimizing label agreement among three annotators (Pustejovsky and Stubbs, 2012; Klie et al., 2024). Figure 1 summarizes our three-staged development process.

We investigate the following research questions:

**RQ1** How effective is a sentence-level classifier in recognizing several text quality classes, given an active learning scenario with a budget of an 8-hour day of annotation?

**RQ2** How does the resulting classifier that has been trained only on few examples perform in comparison to a rule-based approach?

**Contributions** (1) We develop and refine a classification scheme and corresponding guidelines to obtain a gold dataset for evaluation. (2) We perform an active learning experiment investigating the feasibility of a sentence-level classifier, built for a specific domain and language in under 8 hours. (3) We compare our approach to a rule-based one.

**Results** The final classifier shows reasonable performance despite being trained on only 498 examples, reaching macro- and micro-$F_1$ scores of $0.80$ and $0.84$ respectively. Compared to a rule-based baseline, our approach achieves improvements of four to five percentage points in $F_1$ and captures certain types of errors, often more subtle, that the rule-based system tends to miss. We publish the training code, a labeled corpus, and the classifier.

[2]The distinction of what is considered a *small* model is evolving, but the important aspect is that at the current time larger models quickly render computation efforts infeasible, while small models can process large amounts of data despite of limited compute resources.

[3]We rely on the definition of Rogers and Luccioni (2024) for LLMs, which includes encoder models.

[4]We use *model* to refer to the base architecture (e.g., BERT (Devlin et al., 2019)), and *classifier* to denote the model including a task-specific classification head.

## 2 Related Work

**Preprocessing of Web Data** The web has long been used as an important source of text data in natural language processing (Kilgarriff and Grefenstette, 2003), but requires cleaning procedures to remove noisy parts such as boilerplate code, encoding errors, non-linguistic content, or broken text. In the context of LLM training, text cleaning has gained renewed attention, since carefully-curated high-quality data is the currently best known recipe for training strong models (Penedo et al., 2023, 2024). Some preprocessing steps involve handcrafted (often language-specific) rules that have been developed in and adopted from previous work such as the C4 (Raffel et al., 2020) and ROOTS (Laurençon et al., 2022) corpora. Similar (or partly even identical) heuristics have been confirmed in follow-up work and are still used in more recent datasets such as FineWeb (Penedo et al., 2024). Notably, while some preprocessing steps, such as language identification, are realized using models, many cleaning steps still rely on rules.

**Active Learning** Transformer-based language models (Vaswani et al., 2017; Devlin et al., 2019) have shown considerable results in the context of active learning for text classification using only a small amount of data (Ein-Dor et al., 2020; Margatina et al., 2021; Schröder et al., 2022), encouraging this work where a lack of training data is a severe obstacle. With language models continuously increasing in size, some recent approaches even attempt to replace the human annotator with an LLM (Xiao et al., 2023; Kholodna et al., 2024). Many contemporary corpora are, however, very large, and computational costs are still an obstacle for practical active learning (Romberg et al., 2025), therefore we opt to use small language models, which have shown remarkable effectiveness (Nachtegael et al., 2023; Schröder and Heyer, 2024; Gonsior et al., 2025), while at the same time allowing us to process larger volumes of data.

The majority of the recent work at the intersection of language models, active learning, and text classification revolves around single-label classification (among others in the works of Ein-Dor et al. (2020) and Lesci and Vlachos (2024)), while studies focusing on multi-label active learning are rare (e.g., Wertz et al. (2022a,b, 2023) and Wang and Liu (2023)). Moreover, active learning research is often operationalized through simulated experiments (Margatina and Aletras, 2023). Therefore, practical multi-label active learning applications are highly important to investigate the effectiveness of contemporary active learning.

## 3 Quality Criteria and Gold Standard

Our approach is not limited to a specific corpus or language. The following work is conducted at the example of German web text, which is reflected in the class descriptions and textual examples.

### 3.1 Quality Criteria Labels

We define *low-quality* labels to capture visible deficiencies that interrupt the flow of a text on the lexical and syntactical level. Conversely, text without such interruptions is considered *high-quality* (or *clean*). These labels are inspired mostly by rules from related work (e.g., by Raffel et al., 2020; Kreutzer et al., 2022; Laurençon et al., 2022) and from a field-tested rule-based approach, developed for the same kind of data (Goldhahn et al., 2012).[5]

To provide examples to the reader, exemplary sentences with their corresponding label sets are presented in Table 1, where one label is highlighted for each example. The respective classes are defined in the following:

**Sentence Boundary** Sentence boundary errors occur if the start or ending of a sentence is malformed. This is the case if it begins with a lower case letter or an atypical character, or lacks a proper terminal punctuation mark (e.g., period, exclamation mark, or question mark).

**Grammar Mistake** Grammar mistakes are any grammatical errors such as incorrect articles, cases, word order and incorrect use or absence of words. Moreover, random-looking sequences of words, usually series of nouns, should be tagged. In most cases where this label is applicable, the sentence' comprehensibility or message is impaired.

**Spelling Anomaly** A spelling anomaly is tagged when a word does not correspond to German spelling. This includes typos and incorrect capitalization (e.g. "all caps" or lower-case nouns). Spelling anomalies are irregularities that occur within the word boundary, meaning here text between two whitespaces. In particular, individual letters or nonsensical word fragments are also tagged.

[5] `https://github.com/Leipzig-Corpora-Collection/sentencecleaner`

**Punctuation Error** Punctuation errors are tagged if a punctuation symbol has been placed incorrectly or is missing in the intended place. This includes comma errors, missing quotation marks or parentheses, periods instead of question marks or incorrect or missing dashes or hyphens.

**Non-linguistic Content** Non-linguistic content includes all types of encoding errors, language-atypical occurrences of numbers and characters (e.g. random sequences of characters or letters), code (remnants), URLs, hashtags and emoticons.

**Letter Spacing** Letter spacings are deliberately inserted spaces between the characters of a word.

**Clean** Assigned if none of the other labels apply.

### 3.2 Active Learning

To overcome the lack of labeled data, we aim to use active learning (Lewis and Gale, 1994), an iterative approach whose goal is to maximize model performance while minimizing human annotation effort. During each iteration, a so-called *query strategy* selects examples, which are labeled by a human annotator. The model is then retrained on all data labeled so far, and the process repeats in the next iteration.

### 3.3 Gold Standard and Annotation

While this work is not limited to a specific corpus, we need to evaluate the targeted corpus quality classifier. For this reason, we introduce a dataset, which will be used as a gold standard. *This considerable effort is only conducted to enable an experimental evaluation.*

**Data** Through a direct request to the Leipzig Corpora Collection[6] (Goldhahn et al., 2012) we obtained 165 M sentences ($\sim$4 B tokens) of German web text. The resulting text originates from various crawls from 2018. The data is already preprocessed (through text extraction from HTML, sentence splitting, and deduplication). In the following, we operate on the resulting sentences.

**Annotation Process** We rely on agile annotation (Alex et al., 2010; Pustejovsky and Stubbs, 2012; Klie et al., 2024), to iteratively annotate the gold standard over three rounds. During each round, all three annotators (the authors of this work) label a set of given sentences independently. Inter-annotator agreement (IAA) is then assessed using

[6] `https://wortschatz-leipzig.de/en`

| Example sentence | Labels |
|---|---|
| *© zhu difeng \| Visionen zum intelligenten Zuhause gibt es schon lange, und teilweise sind sie sehr ambitioniert.*<br>**EN:** *© zhu difeng \| Visions of the intelligent home have been around for a long time, and some of them are very ambitious.* | Sentence Boundary, Grammar Mistake, Non-linguistic Content |
| *Medisana Luftbefeuchter Ultrabreeze zusätzlichem Nachtlicht*<br>**EN:** *Medisana humidifier Ultrabreeze additional night light* | Grammar Mistake, Sentence Boundary |
| *Wie viel Geld wollen wir fÃ1/4r den Kalender ausgeben?*<br>**EN:** *How much money do we want to spend fÃ1/4r the calendar?* | Spelling Anomaly, Non-linguistic Content |
| *Pegasus Solero SL 28 Zoll 58cm Schwarz..*<br>**EN:** *Pegasus Solero SL 28 Inches 58cm Black..* | Punctuation Error, Sentence Boundary, Grammar Mistake |
| *Zweitens: Ich L I E B E Beeren < 3 In jeglicher Form, Art und GrÃ¶ÃŸe.*<br>**EN:** *Second: I L O V E berries < 3 In all shapes and siÃ¶ÃŸs.* | Non-linguistic Content, Grammar Mistake, Spelling Anomaly, Letter Spacing |
| *V O R T R A G u n d G E S P R Ä C H*<br>**EN:** *T A L K a n d D I S C U S S I O N* | Letter Spacing, Sentence Boundary, Grammar Mistake, Spelling Anomaly |
| *Die Spiel- und Lernstube ist Kontakt- und Anlaufstelle für Kinder, Jugendliche, Eltern und Bewohner im Stadtteil.*<br>**EN:** *The play and learning center is a point of contact and a drop-in center for children, adolescents, parents, and residents in the neighborhood.* | Clean |

Table 1: Exemplary sentences (in German with an English translation below) and their respective gold labels.

Cohen's Kappa for each pair of annotators and each label. The score is analyzed and shortcomings of the guidelines or difficult edge cases are discussed. After this, class definitions or the guidelines are adjusted (e.g., by adding new positive or negative examples) and the sentences are relabeled.

Since we follow an iterative approach, any time we revise the guidelines for *all* classes, we would need to relabel every sentence in the batch to reflect the updated definitions. To keep the effort manageable, we relabeled the entire batch only in the first round. In the subsequent two rounds, we focused on specific classes that showed significant discrepancies between annotators.

The first batch of examples (460 in total) was collected using multi-label Adaptive-Active-Learning (Li and Guo, 2013) to primarily identify error cases. The second batch consisted of 600 randomly selected examples to increase text diversity, while the third batch comprised 275 manually collected examples aimed to cover previously underrepresented classes.

We report agreement scores of each batch of the initially labeled version in comparison with the final version in Table 2. We see the largest improvements in the first batch. This can be attributed to the initially low inter-annotator agreement, which prompted a thorough discussion, followed by a complete relabeling of the whole batch. We repeated this process two times. After that, we only selected examples from low-performing classes. We saw a moderate increase in IAA in the third batch, but not in the second one. Although there were clear improvements in the initial IAA values for batches 2 and 3, the final IAA value of 0.74 for the first batch could not be reached.

| Batch | Initial IAA | Final IAA | Size |
|---|---|---|---|
| First batch | 0.54 | 0.74 | 460 |
| Second batch | 0.71 | 0.71 | 600 |
| Third batch | 0.72 | 0.75 | 275 |

Table 2: IAA (Cohen's Kappa) between three coders for the iterative labeling approach over three iterations.

**Final Dataset** The three batches are combined and a majority voting is used to merge the labels. We had to discard 16 sentences which contained harmful content or Personally Identifiable Infor-

| Label | IAA | # Examples |
|---|---|---|
| Sentence Boundary | 0.86 | 439 |
| Grammar Mistake | 0.76 | 594 |
| Spelling Anomaly | 0.61 | 290 |
| Punctuation Error | 0.41 | 78 |
| Non-linguistic Content | 0.75 | 147 |
| Letter Spacing | 0.96 | 25 |
| Clean | 0.80 | 577 |
| **Avg/Total** | 0.74 | 2150 |

Table 3: Class-wise and averaged inter-annotator agreement, class distribution and number of class examples (2150 labels in 1319 sentences) of our gold standard.

mation. The final inter-annotator agreement and additional dataset statistics are shown in Table 3. According to the Kappa interpretation of Landis and Koch (1977), with an average of 0.74 we reach a substantial agreement level (0.6–0.8).

## 4 Experiments

In this experiment, we examine the feasibility of detecting the proposed text quality classes, in a scenario where training data and annotation time are severely limited (**RQ 1**). For this purpose, we bootstrap a classifier with active learning that is evaluated against the annotated gold standard. To further assess the effectiveness of the resulting classifier, we compare it to a rule-based baseline, which detects similar textual issues (**RQ 2**).

### 4.1 Experimental Setup

To reflect realistic constraints, we simulate the scenario of a small team facing large volumes of unlabeled data with a limited annotation budget by imposing a time budget of one working day (8 hours). Active learning is warm-started with an initial training pool of 70 hand-picked examples ($\sim$ 10 examples per class). In each round, the query strategy returns a batch of 20 sentences to a human annotator. To improve the fine-tuning stability, we train the classifier from scratch, e.g., from the pre-trained base model, after every batch.

**Data** A new dataset is used, which was created as described in Section 3, based on more recent crawling data of the same project, crawled in 2022 with 136 M extracted sentences ($\sim$ 3.4 B tokens).

**Classification** For classification, we use SetFit (Tunstall et al., 2022), an efficient fine-tuning paradigm that leverages contrastive learning. Using a sampling strategy, it generates similar and dissimilar sentence pairs which are used to train a siamese network. In the multi-label setting, sentences are considered similar (positive pair), if they have a label in common, and dissimilar otherwise (negative pair). While there are variations to SetFit, we stayed close to the original version in which a Sentence Transformer (Reimers and Gurevych, 2019) is fine-tuned and the classification operates on the resulting embeddings. Instead of a logistic regression head, however, we opted for a neural network head, which is faster for even a moderate number of classes at a similar classification performance.

**Base Model** As the base model, we create a Sentence Transformer (Reimers and Gurevych, 2019) by mean pooling over the output layer from multilingual DistilBERT (Sanh et al., 2019) (135 M parameters). Compared to BERT, DistilBERT contains only half the number of layers and is therefore more efficient regarding training and inference.

**Query Strategy** We use multi-label Adaptive-Active-Learning (AAL; Li and Guo, 2013) as the query strategy, which balances two scores to find informative samples: (1) Max-Margin Uncertainty Sampling (MMUS) and (2) Label-Cardinality-Inconsistency (LCI). MMUS calculates the distance between the maximum of the predicted negative labels and the minimum of the predicted positive labels, according to a fixed threshold (e.g., 0.5). If the distance is small, the sample is considered highly informative. LCI assumes that multi-label instances often have a similar label count. It computes the deviation of the predicted label count from the average in the so far annotated data (for details, see Section 4 in Li and Guo, 2013).

One limitation of selecting data based on predictions is that the data has to be passed forward through the classifier before any selection criteria can be applied. To make this step feasible, during every round we subsample 10 K unlabeled sentences before applying the query strategy. A batch of the 20 highest-scoring samples is selected.

**Implementation** The implementation for the active learning routine and query strategies are based on `small-text`[7] (Schröder et al., 2023), an active learning library specialized in text classification, with integrations for transformers and SetFit.

[7] https://github.com/webis-de/small-text

| Metric | Value |
|---|---|
| $\mathbf{F}_1^{macro}$ | 0.80 |
| $\mathbf{F}_1^{micro}$ | 0.84 |
| **Subset Acc** | 0.67 |

Table 4: Active learning results for 498 examples.

| Class | $F_1$ | # Count |
|---|---|---|
| Sentence Boundary | 0.96 | 169 |
| Grammar Mistake | 0.86 | 256 |
| Spelling Anomaly | 0.57 | 158 |
| Punctuation Error | 0.62 | 77 |
| Non-linguistic Content | 0.77 | 110 |
| Letter Spacing | 0.94 | 11 |
| Clean | 0.86 | 145 |

Table 5: Class-wise active learning results with the number of training examples per class.

To ease the process for the annotator, we connected the annotation tool `argilla`[8] to our backend.

## 5 Results

### 5.1 Active Learning Experiment

The experiment took 7 hours and 50 minutes in total, during which 22 batches with 20 examples each were processed. Re-training from scratch with every newly annotated batch required overall 5 hours on one Nvidia Tesla A30 (24 GB), querying in total $\sim$ 1 hour, labeling less than 2 hours. The human annotator in this experiment was the first author of this paper. During the annotation process 12 samples had to be discarded due to the problems mentioned above (see Section 3.3).

In Table 4, we report $F_1$ and subset accuracy of the last active learning round on our gold standard, which used 498 examples for the training (428 samples + 70 initial examples). The classifier achieves average scores of $F_1^{macro} = 0.80$ and $F_1^{micro} = 0.84$. The subset accuracy of 0.67 is sufficiently high, considering that only exactly matching label combinations are considered correct. The class scores vary considerably, ranging from 0.57 to 0.96 (see Table 5). Every second sentence was annotated with the label *"Grammar Mistake"* (256 examples), followed by *"Sentence Boundary"* (169 examples) in terms of frequency (last column of Table 5). When comparing $F_1$ values, there is no indication that a higher number of training examples always results in higher scores (e.g., when comparing *"Grammar Mistake"* $= 0.86$ and *"Sentence Boundary"* $= 0.96$). This can also be shown with other low-quality classes, notably including *"Letter Spacing"* that only required 11 examples to achieve a score of 0.94.

To further investigate the active learning process, we reproduce the classifier's progression during the experiment by training checkpoints with different seeds at every two batches of training data and plot the results (see Figure 2). For example, we train five times with all the training data, which was sampled until batch four (70 initial and 80 queried examples), then train five times on batch six, and so on. Figure 2 shows improvements across all classes, with steeper increases initially that gradually level off over the course of the experiment, albeit at different rates. Although the macro $F_1$ curve shows signs of stagnation during the last two batches, increasing the annotation budget may yield further improvements. However, the point at which performance would begin to decline remains unclear. One approach would be to proceed cautiously by reducing the active learning batch size.

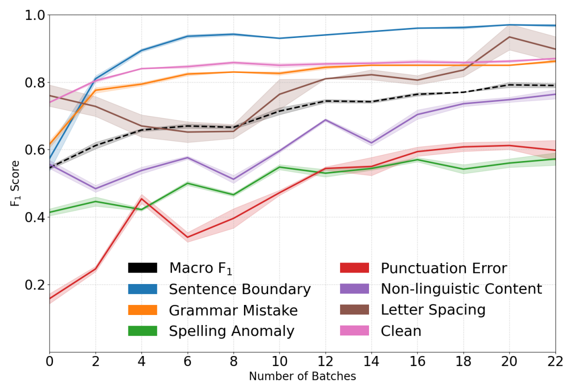



Figure 2: Macro and class-wise $F_1$ in relation to train examples per batch, showing classification progress during the active learning cycle. Re-trained with 5 seeds at every second batch. Batch 0 are the 70 initial examples.

### 5.2 Comparison to Rule-Based Filtering

To further investigate the classifier's performance, we compare it against a rule-based baseline, Sentencecleaner, which was developed within the context of the project through which our dataset was obtained (Goldhahn et al., 2012). This tool applies a set of 40 rules to filter out low-quality sentences and is typically used on web crawling data. These

[8] https://github.com/argilla-io/argilla

| | Not-Clean | Clean |
|---|---|---|
| Sentencecleaner | 0.8181 | 0.8150 |
| Corpus Quality Classifier | **0.8673** | **0.8603** |

Table 6: Comparison in $F_1$ score between our supervised approach and a rule-based baseline, both developed for the same task and dataset. Evaluated against *not-clean* and *clean* sentences from our gold standard.

rules include checks for character or symbol ratios, letter spacing and invalid sentence boundaries.[9]

**Quantitative Comparison** We apply a black-box evaluation, comparing both methods solely based on their outputs against our gold standard, focusing on their ability to detect impaired and clean sentences. Classifications are considered *not-clean* when the classifier identifies any low-quality label or when any Sentencecleaner rule applies, whereas classifications are considered *clean* when the class *"Clean"* is correctly predicted or no Sentencecleaner rules are applicable. Table 6 shows that our approach outperforms by 4.92 (not-clean) and 4.53 (clean) percentage points in $F_1$, demonstrating both better error detection and clean text recognition.

**Qualitative Comparison** To have a better understanding of which *additional* patterns the classifier can find, we perform a brief qualitative comparison.

We first review examples that were incorrectly filtered out by the rule-based baseline, but correctly retained by our approach. Among those 39 sentences, 12 could be captured with simple rule adjustments. This would be feasible, for example, for the rule according to which sentences are marked as not clean if they begin with a number that is not part of a valid date format. In addition, 26 sentences were filtered out by a rule prohibiting a sentence length of more than 255 characters which is a good example of how difficult it is to find reasonable thresholds. Of the 56 sentences that triggered this rule, half were true positives and half were false positives, yielding a precision of just 50%.

We also look at the 107 sentences, which our classifier is correctly predicting as not clean and the rule-based approach missed. Among these, it is notable that the majority (92 sentences) contain *"Grammar Mistake"* in their label set, which covers all sorts of violations that affect the comprehensibility of a sentence. To further investigate the error patterns, we grouped them into different subcategories and briefly describe them (see Table 7). There are 51 cases where a finite verb form is missing (*"Missing Predicate"*), e.g. headlines (news, e-commerce, advertisement, etc.), product descriptions or bullet points. They all have typical characteristics of well-formed sentences, like starting with a capital lettered word, ending with a punctuation mark while not containing any misplaced or random symbols. The second largest group contains 28 cases with foreign language parts ($\sim$50% non-German text), which are, according to our definition, grammar violations (*"Language Mixing"*). The remaining cases comprise various textual anomalies, including incoherent sentences, missing word boundaries causing lexical merging, and sentences that appear truncated (*"Gibberish"*, *"Merged Words"* and *"Truncation"*).

To assess the severity of the overlooked errors, We also examined the classifier's limitations, specifically the 104 sentences it mistakenly identified as clean. When looking at the examples and their gold labels, the two most common label sets are the single labels *"Spelling Anomaly"* (40 cases) and *"Grammar Mistake"* (31 cases). Single-label occurrences often reflect subtle errors, which could be confirmed by examining the actual text content.

## 6 Discussion

Considering the total amount of time (8h) and training data (498 examples), we argue that our proposed setup worked sufficiently well to build a classifier for text cleaning and could serve as a blueprint for data efficient training. Although this has been demonstrated on German web crawls, our pipeline is agnostic to language and domain: only the annotation scheme and seed examples would need to be adapted.

Without further experiments, however, it is not clear how these methods will perform in comparison to traditional supervised-learning using random data points. Nevertheless, during the annotation of the second batch of the gold standard—600 random examples—we observed that $\sim$50% of sentences were clean. In contrast, within the active learning training data, the class *"Clean"* was sampled only 145 times (29%), thereby focusing annotation effort on noisy examples. This suggests that the traditional supervised classifier will likely be trained on fewer error cases compared to using ac-

[9] https://github.com/Leipzig-Corpora-Collection/sentencecleaner

| Pattern | Exemplary sentence |
|---|---|
| Missing Predicate | *Große Abgeschlagenheit und Trägheit des Körpers.*<br>**EN:** *Great fatigue and sluggishness of the body.* |
| Language Mixing | *Nun, das lässt sich übertragen: What is a school but the people?*<br>**EN:** *Well, that can be transferred: What is a school but the people?* |
| Gibberish | *Ist dort Folklore, war schon der 16. Angriff.*<br>**EN:** *Is folklore there, was already the 16th attack.* |
| Merged Words | *Erdäpfelgulasch - Der SpeisenzustellerEs befinden sich keine Produkte im Warenkorb.*<br>**EN:** *Potato goulash - The food delivererThere are no products in your basket.* |
| Truncation | *Wie in anderen Bundesländern muss auch in.*<br>**EN:** *As in other federal states, this must also be done in.* |

Table 7: Various low-quality patterns, the classifier **additionally** found, in comparison to a rule-based approach.

tive learning, which will reduce performance of our low-quality classes, but may improve on *"Clean"*.

When looking at *additional* low-quality patterns that our approach identifies (Table 7), we find various textual problems, some of which are less obvious to recognize by looking at the surface structure alone. One could argue that certain *"Missing Predicate"* instances, such as headlines that only lack finite verbs, do not constitute low-quality text. While this does make sense at the document level, where the text might serve its function as a summarizing heading, our sentence-level approach assesses quality focused on syntactically valid sentences.

The comparison also demonstrates the inherent problem in selecting suitable threshold values in rule-based approaches, as can be seen with the imprecise sentence length heuristic.

It is worth noting that our seven-class schema represents only a *first effort* to define web text quality and does not fully capture what constitutes low (or high) quality sentences. This work focused on data efficient training methods rather than the development of a comprehensive taxonomy.

To obtain a rough estimate of GPU requirements for corpus preprocessing, we processed 1 M sentences ($\sim$ 25 M tokens) on a Nvidia H100 (80GB), which took $\sim$ 123.10 s. Extrapolating to 1 T corresponds to about 1388 GPU hours.[10] While this constitutes a significant resource demand, scaling across multiple GPUs or nodes would render even corpora an order of magnitude larger computationally feasible. Moreover, this estimate reflects the contemporary throughput, but as GPU capabilities and computational speed continue to advance, the boundary of what is feasible will steadily expand.

[10]We use vanilla inference using the SetFit library, but we observed that the throughput plateaued beyond a certain batch size, even though GPU memory was not saturated. We suspect that with code optimizations, the runtime could be further reduced, so the reported number serves as a lower bound.

## 7 Conclusions

In this work, we proposed a labeling scheme for corpus quality classification, provided a gold standard of 1,319 annotated sentences for German web data, and applied active learning to bootstrap a classifier that predicts corpus quality indicators. For evaluation purposes, we created a gold standard using an iterative annotation process, which yielded a corpus with substantial inter-annotator agreement (with a Cohen's Kappa of 0.74), making it suitable for further use.

Using a multi-label active learning setup, we trained a classifier that predicts the defined quality labels for German language with a macro $F_1$ score of 0.80 and micro $F_1$ of 0.84 despite using only 498 training examples in total, labeled over the course of 8 hours. We showed that our supervised approach outperforms a rule-based one developed for the same task. Additionally, we find that the classifier is able to capture error types, particularly those involving the comprehensibility of a sentence, which the rule-based baseline tends to miss.

This work demonstrates a successful proof of concept for enabling model-based filtering through LLM-based active learning for text classification in resource-constrained scenarios. As capabilities of LLMs grow and computational costs decline, preprocessing of larger volumes becomes increasingly feasible, and as a result we predict that preprocessing will shift towards small efficient models, making preprocessing for specific languages and domains increasingly prevalent.

## Limitations

We did not continue training a pre-trained Sentence Transformer (ST) model for SetFit, but bootstrapped one from a vanilla transformer model (due to a lack of a comparable German ST model), which may produce suboptimal sentence-level embeddings compared to one whose representations have been pre-trained on sentence pairs and cosine similarity loss. We encourage exploring the use of a pre-trained ST, as this could further improve performance. While we were not aware of a suitable model for German, the multi-lingual model from Reimers and Gurevych (2019) is a promising candidate for further investigation.

During the creation of the gold standard, we discovered a bug in the query strategy used to select data for the first batch. We assume these sentences were drawn roughly randomly like the second batch, but still covered more error cases.

## Ethical Statement

The dataset annotations may exhibit bias reflecting the perspectives of the annotators, who are all computer science researchers, potentially limiting the diversity of opinions represented in our quality assessments. Additionally, our definition of high quality correlates strongly with standard German grammar, which may inadvertently exclude dialectal variations or linguistic phenomena such as code-switching. This presents a particular concern given that LLM pre-training corpora should ideally be as representative as possible of natural language variation. To address these limitations, we will release our dataset and model to enable further investigation of these problems.


## Acknowledgments

This work has been partially funded by the German Federal Ministry of Research, Technology, and Space (BMFTR) under the grant numbers 01IS24077A and 01IS24037B. This work has been partially supported by the ScaDS.AI Center for Scalable Data Analytics and Artificial Intelligence, project identification number: ScaDS.AI.

Computations for this work were done (in part) using resources of the Leipzig University Computing Center.


## References


Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, et al. 2024. A survey on data selection for language models. *arXiv preprint arXiv:2402.16827*.

Bea Alex, Claire Grover, Rongzhou Shen, and Mijail Kabadjov. 2010. Agile corpus annotation in practice: An overview of manual and automatic annotation of CVs. In *Proceedings of the Fourth Linguistic Annotation Workshop*, pages 29–37, Uppsala, Sweden. Association for Computational Linguistics.

Zachary Ankner, Cody Blakeney, Kartik Sreenivasan, Max Marion, Matthew L. Leavitt, and Mansheej Paul. 2025. Perplexed by perplexity: Perplexity-based data pruning with small reference models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

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.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the Leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Julius Gonsior, Tim Rieß, Anja Reusch, Claudio Hartmann, Maik Thiele, and Wolfgang Lehner. 2025. Survey of active learning hyperparameters: Insights from a large-scale experimental grid.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Erik Henriksson, Otto Tarkka, and Filip Ginter. 2025. FinerWeb-10BT: Refining web data with LLM-based line-level filtering. In *Proceedings of the Joint 25th Nordic Conference on Computational Linguistics and 11th Baltic Conference on Human Language Technologies (NoDaLiDa/Baltic-HLT 2025)*, pages 258–268, Tallinn, Estonia. University of Tartu Library.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov.

2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Nataliia Kholodna, Sahib Julka, Mohammad Khodadadi, Muhammed Nurullah Gumus, and Michael Granitzer. 2024. Llms in the loop: Leveraging large language model annotations for active learning in low-resource languages. In *Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Proceedings, Part X*, page 397–412, Berlin, Heidelberg. Springer-Verlag.

Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29(3):333–348.

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2024. Analyzing dataset annotation quality management in the wild. *Computational Linguistics*, 50(3):817–866.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, Giada Pistilli, Olivier Nguyen, Somaieh Nikpoor, Maraim Masoud, Pierre Colombo, Javier de la Rosa, Paulo Villegas, Tristan Thrush, Shayne Longpre, Sebastian Nagel, Leon Weber, Manuel Muñoz, Jian Zhu, Daniel Van Strien, Zaid Alyafeai, Khalid Almubarak, Minh Chien Vu, Itziar Gonzalez-Dios, Aitor Soroa, Kyle Lo, Manan Dey, Pedro Ortiz Suarez, Aaron Gokaslan, Shamik Bose, David Adelani, Long Phan, Hieu Tran, Ian Yu, Suhas Pai, Jenny Chim, Violette Lepercq, Suzana Ilic, Margaret Mitchell, Sasha Alexandra Luccioni, and Yacine Jernite. 2022. The bigscience roots corpus: A 1.6tb composite multilingual dataset. In *Advances in Neural Information Processing Systems*, volume 35, pages 31809–31826. Curran Associates, Inc.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, et al. 2022. The bigscience roots corpus: A 1.6 tb composite multilingual dataset. *Advances in Neural Information Processing Systems*, 35:31809–31826.

Pietro Lesci and Andreas Vlachos. 2024. AnchorAL: Computationally efficient active learning for large and imbalanced datasets. 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 8445–8464, Mexico City, Mexico. Association for Computational Linguistics.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12. Springer, ACM/Springer.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. 2024. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282.

Xin Li and Yuhong Guo. 2013. Active learning with multi-label SVM classification. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 1479–1485. IJCAI/AAAI.

Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. 2024. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. 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 3245–3276, Mexico City, Mexico. Association for Computational Linguistics.

Katerina Margatina and Nikolaos Aletras. 2023. On the limitations of simulating active learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4402–4419, Toronto, Canada. Association for Computational Linguistics.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Charlotte Nachtegael, Jacopo De Stefani, and Tom Lenaerts. 2023. A study of deep active learning methods to reduce labelling efforts in biomedical relation extraction. *PLOS ONE*, 18(12):1–23.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Hamza Alobeidli, Alessandro Cappelli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. The RefinedWeb dataset for falcon LLM: outperforming curated corpora with web data only. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

James Pustejovsky and Amber Stubbs. 2012. *Natural Language Annotation for Machine Learning: A Guide to Corpus-Building for Applications*. O'Reilly Media.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Anna Rogers and Sasha Luccioni. 2024. Position: Key claims in LLM research have a long tail of footnotes. In *Forty-first International Conference on Machine Learning*.

Julia Romberg, Christopher Schröder, Julius Gonsior, Katrin Tomanek, and Fredrik Olsson. 2025. Have LLMs made active learning obsolete? Surveying the NLP community. *arXiv preprint arXiv:2503.09701*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Christopher Schröder and Gerhard Heyer. 2024. Self-training for sample-efficient active learning for text classification with pre-trained language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11987–12004, Miami, Florida, USA. Association for Computational Linguistics.

Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. 2023. Small-text: Active learning for text classification in python. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 84–95, Dubrovnik, Croatia. Association for Computational Linguistics.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Tristan Thrush, Christopher Potts, and Tatsunori Hashimoto. 2025. Improving pretraining data using perplexity correlations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv preprint arXiv:2209.11055*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Mengqi Wang and Ming Liu. 2023. An empirical study on active learning for multi-label text classification. In *Proceedings of the Fourth Workshop on Insights from Negative Results in NLP*, pages 94–102, Dubrovnik, Croatia. Association for Computational Linguistics.

Lukas Wertz, Jasmina Bogojeska, Katsiaryna Mirylenka, and Jonas Kuhn. 2022a. Evaluating pre-trained sentence-BERT with class embeddings in active learning for multi-label text classification. 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 2: Short Papers)*, pages 366–372, Online only. Association for Computational Linguistics.

Lukas Wertz, Jasmina Bogojeska, Katsiaryna Mirylenka, and Jonas Kuhn. 2023. Reinforced active learning for low-resource, domain-specific, multi-label text classification. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10959–10977, Toronto, Canada. Association for Computational Linguistics.

Lukas Wertz, Katsiaryna Mirylenka, Jonas Kuhn, and Jasmina Bogojeska. 2022b. Investigating active learning sampling strategies for extreme multi label text classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4597–4605, Marseille, France. European Language Resources Association.

Alexander Wettig, Aatmik Gupta, Saumya Malik, and Danqi Chen. 2024. Qurating: Selecting high-quality data for training language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 52915–52971. PMLR.

Ruixuan Xiao, Yiwen Dong, Junbo Zhao, Runze Wu, Minmin Lin, Gang Chen, and Haobo Wang. 2023. FreeAL: Towards human-free active learning in the era of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14520–14535, Singapore. Association for Computational Linguistics.

# Fine-Grained Arabic Offensive Language Classification with Taxonomy, Sentiment, and Emotions

**Natalia Vanetik** [1]
natalyav@sce.ac.il

**Marina Litvak** [1]
marinal@sce.ac.il

**Chaya Liebeskind** [2]
liebchaya@gmail.com

[1]Shamoon College of Engineering, Beer-Sheva, Israel
[2]Jerusalem College of Technology, Jerusalem, Israel

## Abstract

Offensive language detection in Arabic is a challenging task because of the unique linguistic and cultural characteristics of the Arabic language. This study introduces a high-quality annotated dataset for classifying offensive language in Arabic, based on a structured taxonomy, categorizing offensive content across seven levels, capturing both explicit and implicit expressions. Utilizing this taxonomy, we re-annotate the FARAD-500 dataset, creating reFarad-500, which provides fine-grained labels for offensive texts in Arabic. A thorough dataset analysis reveals key patterns in offensive language distribution, emphasizing the importance of target type, offense severity, and linguistic structures. Additionally, we assess text classification techniques to evaluate the dataset's effectiveness, exploring the impact of sentiment analysis and emotion detection on classification performance. Our findings highlight the complexity of Arabic offensive language and underscore the necessity of extensive annotation frameworks for accurate detection. This paper advances Arabic natural language processing (NLP) in resource-constrained settings by enhancing the recognition of hate speech and fostering a deeper understanding of the linguistic and emotional dimensions of offensive language.

## 1 Introduction

Arabic offensive language detection is a crucial but difficult undertaking that necessitates a thorough comprehension of both linguistic structures and cultural context. Direct insults and hate speech are examples of explicit offensive content. Implicit offensive content necessitates a more thorough contextual study to determine its intent. Even though Arabic language processing has improved due to recent developments in natural language processing (NLP), current categorization frameworks frequently lack the granularity required to effectively capture objectionable statements in various Arabic dialects.

We present a high-quality data set that is annotated in accordance with the ArSOL taxonomy (Liebeskind et al., 2024), but unlike the original ArSOL work, we reapply it with stricter guidelines and expanded multi-label capability, to overcome these issues. It comprises seven hierarchical levels, distinguishing between explicit and implicit offenses, and categorizing offensive content based on target presence, vulgarity, offense severity, and type.

As part of this study, we start from the FARAD-500 dataset, which aggregates Arabic offensive language texts from multiple sources, but our work departs from it by systematically correcting annotation inconsistencies. The original FARAD-500 dataset provides valuable offensive language examples, but several issues limit its utility: inconsistent application of labels, overuse of vague categories, and lack of multi-label annotations for complex instances. Our re-annotation aims to address these issues by applying the ArSOL taxonomy rigorously and by instructing annotators to distinguish between overlapping categories when relevant. This effort resulted in reFarad-500, a dataset that enhances classification precision across different offensive categories. We analyze the dataset using various NLP techniques, including sentiment analysis and emotion detection, to explore their role in improving offensive language classification.

Through an extensive evaluation pipeline, we assess the quality and utility of the dataset by training text classification models on it and evaluating their performance using standard metrics. We analyze results across multiple annotation levels to examine how the structure of the annotation scheme impacts classification performance. Our results demonstrate the advantages of a structured annotation approach and offer important insights into the



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 110–119, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-013

patterns of offensive language in Arabic.

We believe that incorporating sentiment analysis and emotion detection can provide additional information about the speaker's emotional context, even though the majority of previous work has been on explicitly recognizing offensive words. If one is aware of the emotions that accompany offensive remarks, it would be possible to categorize different levels of offensiveness more accurately (Mnassri et al., 2023b).

By concentrating on Arabic, this work helps close the gap in natural language processing for medium- and low-resource languages. Despite being extensively spoken, Arabic, a morphologically rich language, is nevertheless underrepresented in high-quality annotated datasets for offensive material. Our method tackles important issues like the lack of data, consistent annotations, and the intricate relationship between social context and linguistic structure. We offer a fine-grained, reusable resource and experimental approach that is applicable to other languages with comparable limitations by re-annotating an existing dataset and incorporating sentiment and emotion features.

## 2 Related Work

Various taxonomies classify offensive language at different levels. The term "offensive language" has been defined in diverse ways in prior research; in this paper, we adopt a broad definition stating that offensive language is any form of communication that intentionally or unintentionally conveys hostility, disrespect, or harm toward individuals or groups.

Works of Zampieri et al. (2019a,b) classify content as offensive or not, then as targeted insults or threats, and finally by target type (groups, individuals, etc.). The Nexus Linguarum Working Group (Lewandowska-Tomaszczyk et al., 2021) defined offensive and non-offensive language, targeted and non-targeted insults, and explicit versus implicit language with two primary levels and four sub-levels. Lewandowska-Tomaszczyk et al. (2022) tested a schema for explicit and implicit language and proposed a simplified, unified approach with direct and implied offensiveness in (Lewandowska-Tomaszczyk, 2023). The authors demonstrate that the SOL taxonomy helps identify offensive language in English by showing that its categories align with semantic patterns in word embeddings and yield consistent annotations with high inter-annotator agreement. Liebeskind et al. (2023) have shown that this taxonomy can be successfully applied to Hebrew.

Despite their differences in granularity and structure, these taxonomies all aim to formalize the idea of offensive language. The definition of offensive content is still debatable and complex, though. Related concepts including hate speech, toxicity, abusive language, and incivility have been used in earlier research; meanings range from overtly hostile utterances to more subdued expressions like sarcasm, stereotyping, or exclusionary discourse. In this work, we adopt a more expansive conceptualization that acknowledges sociolinguistic variation in the expression and perception of offense, particularly in morphologically rich and culturally diverse languages like Arabic, and that takes into account both explicit and implicit forms of offense.

To formalize this view, we rely on a structured taxonomy introduced in (Liebeskind et al., 2024) provides a comprehensive framework for categorizing Arabic offensive language. To simplify addressing it in the paper, we denote it by the ArSOL taxonomy. This seven-level taxonomy refines and extends the framework proposed by Lewandowska-Tomaszczyk et al. (2023) and builds on Zampieri et al. (2019a,b) to capture both explicit and implicit offensive language. The taxonomy categorizes offensive language into seven levels: Levels 1 to 6 focus on explicit categories, while Level 7 addresses implicit language. In this study, we focused on Levels 1–6 due to data limitations. Level 7 will be addressed in future extensions. Figure 1 depicts levels 1-6 of ArSOL with English translations.

Multiple datasets for offensive language detection in Arabic have been introduced, reflecting the linguistic and cultural diversity of Arab-speaking regions. Early datasets focused on specific hate speech types: for example, Albadi et al. (2018) contains texts targeting religious prejudice, while Aref et al. (2020) created a dataset focused on religious hate speech concerning the Sunni-Shia divide. Expanding thematic scope, Mulki and Ghanem (2021) developed the Let-Mi dataset, which provides versatile examples of misogynistic behavior.

Other datasets use multidimensional annotation frameworks to capture complex phenomena. Ousidhoum et al. (2019) presented a multilingual dataset annotated for hostility, directness, and target attributes such as religion or sexual orientation. Similarly, Ahmad et al. (2024) released a multi-class dataset of tweets categorized into four sentiment-based hate speech classes.

Researchers have also explored platform-specific data sources, including YouTube comments (Alakrot et al., 2018) and news articles (Chowdhury et al., 2020; Mubarak et al., 2017). Furthermore, several studies address offensive language in different Arabic dialects (Mulki et al., 2019; Haddad et al., 2019; Mubarak et al., 2020; Litvak et al., 2021; Essefar et al., 2023; Alhazmi, 2023). The FARAD-500 dataset, proposed by Liebeskind et al. (2024), focuses on Modern Standard Arabic (MSA) and Levantine dialects and contains 500 offensive texts annotated according to the ArSOL taxonomy.

Our work complements and extends prior efforts by re-annotating the FARAD-500 dataset to improve annotation accuracy and balance across offense categories. The refined dataset, reFarad-500, ensures a more balanced representation of offensive language types, facilitating improved model training and evaluation. We also evaluate the effectiveness of the ArSOL taxonomy by training text classification models on the refined dataset. In addition, we investigate how sentiment analysis and emotion detection can assist offensive language detection. Lastly, the re-annotated dataset is made publicly available to support further advances in Arabic NLP and offensive language identification. Although extensive research exists on Arabic offensive language detection, few studies explore integrating sentiment and emotion analysis to enhance classification. We investigated their role in enhancing offensive language classification since previous work (Plaza-del Arco et al., 2021; Mnassri et al., 2023a; Samghabadi et al., 2020; Elmadany et al., 2020; Althobaiti, 2022) shows the advantages of combining these approaches.

# 3 The reFarad-500 Dataset

## 3.1 Data Preprocessing

We used the FARAD-500 dataset of (Liebeskind et al., 2024) as a starting point, but our work substantially revises and extends it. FARAD-500 was generated from 16 existing Arabic offensive language datasets, ensuring alignment with ArSOL taxonomy. Table 1 lists the datasets and specifies taxonomy levels for which the data is originally annotated (cases, where not all options of a taxonomy level are used, are marked with an asterisk). Most datasets are annotated at level 1 (offensive or not) and partially at level 5 (offense strength), primarily focusing on hate speech. Other taxonomy levels, such as target presence and offense aspects, lack annotation. Furthermore, because of style variations, the dataset only contains texts from Facebook and Twitter, leaving out sources like YouTube and news articles. However, FARAD-500's partial and sometimes inaccurate taxonomy coverage limits its suitability for fine-grained ArSOL-based classification.

| Paper | Source | Levels |
|---|---|---|
| (Albadi et al., 2018) | Twitter | 5* |
| (Ousidhoum et al., 2019) | Twitter | 1, 2, 3, 5* |
| (Mulki et al., 2019) | Twitter | 1, 5* |
| (Zampieri et al., 2020) | Twitter | 1, 4, 5* |
| (Mubarak et al., 2017) | Twitter | 1, 5* |
| (Aref et al., 2020) | Twitter | 5* |
| (Ahmad et al., 2024) | Twitter | 1, 5* |
| (Mulki and Ghanem, 2021) | Twitter | 1, 5*, 6* |
| (Litvak et al., 2021) | Twitter | 1 |
| (Alhazmi, 2023) | Twitter | 1 |

Table 1: Data sources of FARAD-500 (* indicates partial annotation).

## 3.2 Re-annotated dataset reFarad-500

The original FARAD-500 annotations contain partial category coverage and misclassifications, reducing consistency with the ArSOL taxonomy (Liebeskind et al., 2024). We therefore re-annotated the dataset using explicit criteria: correcting label misapplications, ensuring full category coverage, and allowing multi-label assignment when multiple aspects occur in a text. In multiple cases, texts that clearly met the criteria for some of the labels were either misclassified or left unlabeled. To enhance classification accuracy and consistency, this dataset underwent a meticulous re-annotation process to address annotation errors and ambiguities with the help of native Arabic speakers fluent in Modern Standard Arabic (MSA) and Levantine Arabic. The annotators were provided with comprehensive instructions and examples. The main errors we strive to fix were a lack of attention to several aspects expressed in one text, and the use of Other aspect in an erroneous way when other aspects are present in the text. We denote the resulting dataset by reFarad-500. We guided the annotators to mark the aspect as Other only if no other aspect is applicable. Additionally, at level 5, we instructed the annotators to mark each text as either Hate speech or Insult, and then annotate it separately as Threat and as Discredit if necessary to allow for multi-label annotation. We did it to capture the fact that a single offensive text can simultaneously serve multiple

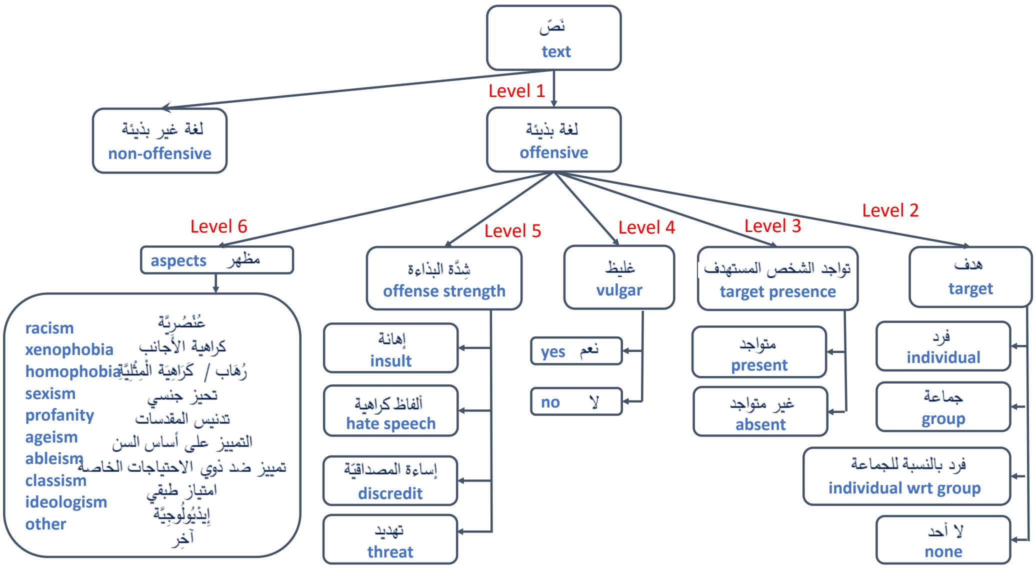



Figure 1: ArSOL taxonomy. Levels 1 through 6 include classifications such as non-offensive vs. offensive (Level 1), the target of the offense (Level 2), the presence or absence of the target (Level 3), vulgar vs. non-vulgar language (Level 4), the severity of the offense (Level 5), and specific types of explicit offenses (Level 6).

functions, such as insulting an individual while also discrediting a group or issuing a threat.

The re-annotation process involved three native Arabic speakers with academic backgrounds in linguistics and NLP. We measured annotator agreement throughout the process using Cohen's Kappa on the full dataset (Table 2). We report Cohen's Kappa coefficient (Cohen, 1960) on the entire dataset of 500 texts. Cohen's Kappa is a statistical measure used to assess the degree of agreement between two raters or annotators when dealing with categorical or ordinal data. The highest level of agreement was found in the Ideologism aspect of level 6, indicating consistency in identifying religiously offensive content. Good to excellent agreement was also noted for Sexism, Xenophobia, and Religion. However, agreement levels were lower

| level | name | kappa |
|---|---|---|
| 2 | target | 0.931 |
| 3 | target presence | 0.775 |
| 4 | vulgar | 0.881 |
| 5 | offense strength (avg) | 0.926 |
| 6 | racism | 0.725 |
| 6 | xenophobia | 0.714 |
| 6 | homophobia | 0.203 |
| 6 | sexism | 0.798 |
| 6 | other | 0.251 |
| 6 | ageism | 1.000 |
| 6 | ableism | 0.127 |
| 6 | classism | 0.322 |
| 6 | ideologism | 0.824 |
| 6 | religion | 0.755 |

Table 2: Annotation agreement for the reFarad-500 dataset.

for some taxonomy aspects, such as Homophobia and Other, implying difficulties in discovering the intent behind the offense in these specific cases. Still, these agreement values are superior to those reported in (Liebeskind et al., 2024), thus justifying the need for re-annotation. In Table 3 we show the agreement between original and new annotations for levels present in both datasets. We see that the aspects most affected by re-annotation are Homophobia", Ableism, and Other, while no changes were made for target presence and vulgarity levels of the taxonomy. Two examples of texts where the annotation was modified appear in Table 4.

| level name | kappa |
|---|---|
| target | 0.996 |
| target presence | 1.000 |
| vulgar | 1.000 |
| racism | 0.895 |
| xenophobia | 0.846 |
| homophobia | 0.203 |
| sexism | 0.817 |
| other | 0.259 |
| ageism | 1.000 |
| ableism | 0.160 |
| classism | 0.406 |
| ideologism | 0.863 |
| religion | 0.788 |

Table 3: Kappa agreement values for FARAD-500 and reFarad-500 datasets.

| text | translation | aspect |
|---|---|---|
| صح لسانك بس الكلب وفي وهذا خنزير نتن | Well said, but the dog is loyal, and this one is a filthy pig. | ableism |
| خال يشق طيز زوجه ديوث.. | An uncle who f**s his wife's a**—he's a cuckold. | homophobia |

Table 4: Examples of re-annotated texts.

## 4 Dataset Analysis

### 4.1 Sentiment Analysis

Sentiment analysis (SA), which helps find sentiment patterns that might be associated with offensive expressions, is an essential tool for text analysis, especially when it comes to identifying offensive language. In offensive language detection, SA can provide additional context by distinguishing between neutral, aggressive, and harmful content, offering a better understanding of intent. For this purpose, we considered two state-of-the-art Arabic pre-trained transformer models: camelBERT (Inoue et al., 2021) and araBERT (Antoun et al., 2020). camelBERT is a state-of-the-art transformer-based model for Arabic that was pretrained on diverse Arabic corpora across dialects and Modern Standard Arabic (MSA), and fine-tuned for various downstream tasks, including sentiment analysis, achieving high accuracy on benchmark datasets ASTD (Nabil et al., 2015) and LABR (Aly and Atiya, 2013). araBERT, similarly, was pretrained on over 200 million Arabic sentences and fine-tuned on multiple sentiment analysis benchmarks, consistently outperforming earlier models and establishing itself as a strong baseline for Arabic NLP tasks.

We applied both camelBERT and araBERT models to the reFarad-500 dataset for sentiment classification. The results, presented in Table 5, show that araBERT produced better quality predictions with more negative labels, while camelBERT assigned a neutral sentiment label to the vast majority of texts, which is surprising. Therefore, we selected araBERT for further analysis. The sentiment distribution over categories of level 6 of the ArSOL taxonomy is depicted in Figure 2.

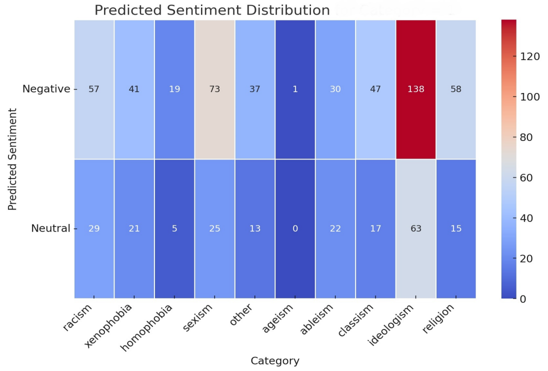



Figure 2: Sentiments seen at level 6 of ArSOL taxonomy.

### 4.2 Emotion Analysis

We also want to investigate whether incorporating emotion detection into offensive language classification will lead to more accurate classification results. Research has indicated that the use of emotional characteristics enhances the identification of hate speech (Plaza-del Arco et al., 2021).

We tested three emotion detection models for Arabic: (1) `hatemnoaman/bert-base-arabic-finetuned-emotion` model (Noaman, 2023) for emotion detection in Arabic which was fine-tuned from `asafaya/bert-base-arabic` on the Emotone dataset (Al-Khatib and El-Beltagy, 2018) and uses eight emotion labels (anger, disgust, fear, joy, neutral, sadness, surprise, trust); (2) `kiroloskhela/Sentiment-Bert` model (Khela, 2023) trained on Emotone (Al-Khatib and El-Beltagy, 2018) and the SetFit/Emotion (Tunstall et al., 2022) datasets with five emotions (disgust, joy, anger, fear, sadness; and (3) araBERT fine-tuned the Arabic Emotion Dataset (Almahdawi and Teahan, 2019) with five emotions (amused, confident, disgust, empathetic, fear). Distribution of detected emotions is shown in Table 6. Because the first two models assign almost all texts the Disgust label, we have elected to proceed with the third model (fine-tuned araBERT). By including emotion recognition in fine-grained offensive language classification, we hope for a better understanding of the intent and severity of offensive text.

### 4.3 Offensive Language Classification

To evaluate the effectiveness of the simplified taxonomy, we performed the classification of texts in the reFarad-500 dataset for every taxonomy level. We also study the effect of Sentiment Analysis (SA) and Emotion Detection (ED) on classification accuracy by using their output for classification. We first generate text representations for the original texts in Arabic, then optionally enhance them with

| Model | Neutral | Positive | Negative |
|---|---|---|---|
| camelBERT | 490 | 5 | 5 |
| araBERT | 269 | 0 | 231 |

Table 5: Sentiment classification results on the reFarad-500 dataset using two Arabic BERT models.

| model | num of texts per emotion |
|---|---|
| asafaya/bert-base-arabic | disgust(448), joy(19), surprise(14),, neutral(7), sadness(5),fear(4), trust(3) |
| kiroloskhela/Sentiment-Bert | disgust(480), joy(10), anger(6), fear(2), sadness(2) |
| fine-tuned araBERT | amused(9), confident(378), disgust (82), empathetic(12), fear(19) |

Table 6: Distribution of detected emotions.

SA or ED output, and split them into training and test sets with the 80%/20% split ratio. Then we apply classification models and report average results (precision, recall, F1 measure, and accuracy). The pipeline of our approach is shown in Figure 3. We only considered categories with at least 10 samples in a minority class, which excluded aspects such as Homophobia and Ageism.

#### 4.3.1 Text Representations and Models

For the offensive language classification, we used three text representations: word n-grams of sizes 1 to 3 (denoted by n-grams in Table 7), tf-idf vectors, and BERT sentence embeddings (denoted by SE). We used two SE models – the Arabic bert-base-arabertv02 model denoted by the araBERT SE model (Antoun et al., 2020) and the multilingual bert-base-multilingual-cased model denoted by mlBERT SE (Devlin et al., 2019). We also investigated the enhancement of these representations with SA and ED labels.

We have applied eXtreme Gradient Boost (XGB) (Chen, 2015), Random Forest (RF) (Pal, 2005), and Logistic Regression (LR) (Kleinbaum et al., 2002) classifiers to these text representations. As baselines, we also applied mlBERT and arBERT and fine-tuned them on the training part of the data.

#### 4.3.2 Results

Table 7 contains the results of classification for levels 2-6 of the taxonomy, showing the difference in performance on syntactic text representations versus semantic representations. At level 6, we classified each offensive aspect separately. In every case, we report the results of the representation-classifier combo that achieved the best accuracy for categories with 10 or more texts in a minority class. Semantic representations (arBERT SE, mlBERT SE) generally outperform syntactic representations (n-grams, tf-idf) in both accuracy and F1-score, highlighting the advantage of contextual embeddings. We can observe that all traditional models consistently outperform both BERT baselines, as can be seen in Table 8 (note some classes were omitted because they had less than 10 texts in the minority class as required by BERT models). We also observed that mlBERT performed better than arBERT in most cases.

Table 10 contains the results of the evaluation of reFarad-500 data enhanced with sentiments predicted by the model described in Section 4.1; the prediction was performed with a train/test split of 80%/20%. The accuracy values in this table indicate that adding SA had minimal impact on most categories, with some slight improvements (Vulgar and Ideologism) but also some decreases (Religion). Representation-wise, n-grams performed consistently well across multiple categories, often achieving competitive or higher accuracy compared to semantic representations like arBERT and mlBERT sentence embeddings. Classifier-wise, XGBoost (XGB) remained the best-performing model.

Table 9 contains the results of the evaluation of reFarad-500 data enhanced with emotions predicted by the model described in Section 4.2; the prediction was performed with train/test split of 80%/20%. The results indicate that incorporating emotion detection (ED) slightly improved accuracy in most categories, particularly in Vulgar and Offense strength, suggesting that emotions contribute to better classification of offensive content intensity. However, for some categories like Target presence and Ideologism, the accuracy changes were minimal, implying that ED might not significantly affect these aspects of offensive language. Overall, the results indicate the potential of emotion-aware models in enhancing fine-grained offensive language classification. Table 11 shows classification results for the reFarad-500 data enhanced by both sentiment analysis (SA) and emotion detection (ED) data. In no case were the results better than the results for the data enhanced only by SA or only by ED.

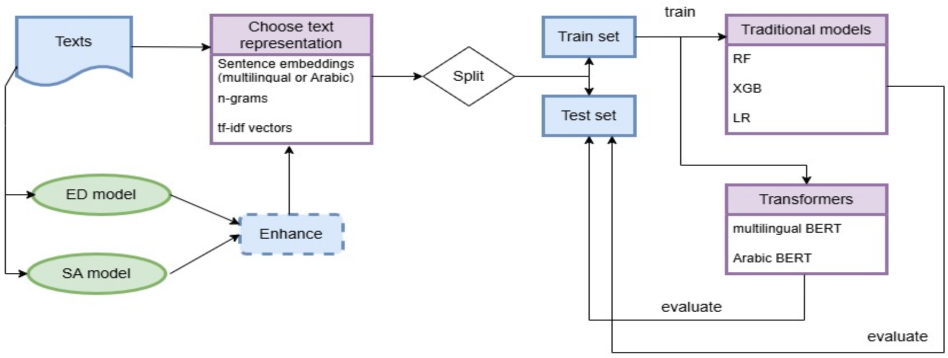



Figure 3: Offensive language classification pipeline.

| level | semantic text representations | | | | syntactic text representations | | | |
|---|---|---|---|---|---|---|---|---|
| | repr. | model | F1 | acc | repr. | model | F1 | acc |
| (2) target | arBERT SE | LR | 0.5802 | 0.7300 | n-grams | XGB | 0.5019 | 0.6800 |
| (3) target presence | mlBERT SE | XGB | 0.5921 | 0.7200 | n-grams | XGB | 0.6685 | 0.7800 |
| (4) vulgar | arBERT SE | RF | 0.5739 | 0.8200 | n-grams | LR | 0.6755 | 0.8300 |
| (5) hate speech/insult | tf-idf | LR | 0.5128 | 0.6300 | arBERT SE | RF | 0.7460 | 0.7500 |
| (5) discredit | mlBERT SE | LR | 0.7288 | 0.7300 | n-grams | XGB | 0.7527 | 0.7600 |
| (6) racism | arBERT SE | XGB | 0.5765 | 0.7900 | n-grams | RF | 0.4350 | 0.7700 |
| (6) xenophobia | mlBERT SE | LR | 0.5821 | 0.8700 | n-grams | RF | 0.4624 | 0.8600 |
| (6) homophobia | arBERT SE | XGB | 0.4872 | 0.9500 | n-grams | RF | 0.4872 | 0.9500 |
| (6) sexism | arBERT SE | XGB | 0.6678 | 0.8400 | n-grams | XGB | 0.6114 | 0.8300 |
| (6) other | arBERT SE | XGB | 0.6094 | 0.9400 | n-grams | XGB | 0.6633 | 0.9300 |
| (6) ageism | arBERT SE | XGB | 1.0000 | 1.0000 | n-grams | XGB | 1.0000 | 1.0000 |
| (6) ableism | arBERT SE | RF | 0.4681 | 0.8800 | tf-idf | XGB | 0.5392 | 0.8800 |
| (6) classism | arBERT SE | RF | 0.4475 | 0.8100 | n-grams | RF | 0.4475 | 0.8100 |
| (6) ideologism | mlBERT SE | LR | 0.6484 | 0.6800 | tf-idf | XGB | 0.6239 | 0.6700 |
| (6) religion | arBERT SE | XGB | 0.5924 | 0.8800 | tf-idf | XGB | 0.6801 | 0.8900 |

Table 7: Comparison of classification results with semantic and syntactic text representations across taxonomy levels (best scores marked in gray).

| level | model | F1 | acc | model | F1 | acc |
|---|---|---|---|---|---|---|
| (2) target | arBERT | 0.3133 | 0.3780 | mlBERT | 0.2510 | 0.4180 |
| (3) target presence | arBERT | 0.5258 | 0.5480 | mlBERT | 0.5738 | 0.6060 |
| (4) vulgar | arBERT | 0.6576 | 0.7180 | mlBERT | 0.4576 | 0.6480 |
| (5) hate speech/insult | arBERT | 0.6094 | 0.6400 | mlBERT | 0.4781 | 0.4800 |
| (5) discredit | arBERT | 0.5478 | 0.5500 | mlBERT | 0.4916 | 0.5100 |
| (6) racism | arBERT | 0.5530 | 0.6580 | mlBERT | 0.3764 | 0.5220 |
| (6) xenophobia | arBERT | 0.4191 | 0.6420 | mlBERT | 0.5003 | 0.8480 |
| (6) homophobia | arBERT | 0.5589 | 0.9140 | mlBERT | 0.5146 | 0.9460 |
| (6) sexism | arBERT | 0.6695 | 0.7740 | mlBERT | 0.4820 | 0.6660 |
| (6) religion | arBERT | 0.5973 | 0.7360 | mlBERT | 0.4607 | 0.6240 |
| (6) ableism | arBERT | 0.5524 | 0.7260 | mlBERT | 0.4480 | 0.7300 |
| (6) classism | arBERT | 0.3826 | 0.5180 | mlBERT | 0.4950 | 0.7920 |
| (6) ideologism | arBERT | 0.7023 | 0.7140 | mlBERT | 0.5221 | 0.5340 |
| (6) other | arBERT | 0.5926 | 0.7660 | mlBERT | 0.5505 | 0.7820 |

Table 8: Fine-tuning of BERT models (best scores are marked in gray).

| level | no ED | | | | with ED | | | |
|---|---|---|---|---|---|---|---|---|
| | repr. | model | F1 | acc | repr. | model | F1 | acc |
| (2) target | arBERT SE | XGB | 0.4887 | 0.7300 | arBERT SE | XGB | 0.4887 | 0.7300 |
| (3) target presence | n-grams | XGB | 0.6685 | 0.7800 | n-grams | XGB | 0.6593 | 0.7700 |
| (4) vulgar | n-grams | LR | 0.6755 | 0.8300 | n-grams | LR | 0.7137 | 0.8500 |
| (5) hate speech/insult | arBERT SE | XGB | 0.7052 | 0.7300 | arBERT SE | RF | 0.6614 | 0.7300 |
| (5) discredit | ngrams | XGB | 0.7029 | 0.7200 | ngrams | XGB | 0.7029 | 0.7200 |
| (5) threat | arBERT SE | RF | 0.4975 | 0.9900 | arBERT SE | RF | 0.4975 | 0.9900 |
| (6) racism | arBERT SE | XGB | 0.5765 | 0.7900 | mlBERT SE | XGB | 0.5847 | 0.8000 |
| (6) xenophobia | mlBERT SE | LR | 0.5821 | 0.8700 | mlBERT SE | LR | 0.5821 | 0.8700 |
| (6) sexism | arBERT SE | XGB | 0.6678 | 0.8400 | arBERT SE | XGB | 0.6678 | 0.8400 |
| (6) religion | arBERT SE | XGB | 0.6094 | 0.9400 | n-grams | XGB | 0.6842 | 0.9400 |
| (6) ableism | arBERT SE | RF | 0.4681 | 0.8800 | arBERT SE | RF | 0.4681 | 0.8800 |
| (6) classism | arBERT SE | RF | 0.4475 | 0.8100 | arBERT SE | RF | 0.4475 | 0.8100 |
| (6) ideologism | mlBERT SE | LR | 0.6484 | 0.6800 | mlBERT SE | XGB | 0.6494 | 0.7000 |
| (6) other | n-grams | XGB | 0.6464 | 0.8900 | n-grams | LR | 0.6464 | 0.8900 |

Table 9: Performance comparison of offensive language classification with and without emotion detection (ED) on the reFarad-500 dataset (best scores are marked in gray).

| level | no SA | | | | with SA | | | |
|---|---|---|---|---|---|---|---|---|
| | repr. | model | F1 | acc | repr. | model | F1 | acc |
| (2) target | arBERT SE | XGB | 0.4887 | 0.7300 | arBERT SE | LR | 0.5802 | 0.7300 |
| (3) target presence | n-grams | XGB | 0.6685 | 0.7800 | n-grams | XGB | 0.6685 | 0.7800 |
| (4) vulgar | n-grams | LR | 0.6755 | 0.8300 | n-grams | LR | 0.6863 | 0.8400 |
| (5) hate speech/insult | arBERT SE | XGB | 0.7052 | 0.7300 | arBERT SE | XGB | 0.6881 | 0.7200 |
| (5) discredit | ngrams | XGB | 0.7029 | 0.7200 | ngrams | XGB | 0.7220 | 0.7300 |
| (5) threat | arBERT SE | RF | 0.4975 | 0.9900 | arBERT SE | RF | 0.4975 | 0.9900 |
| (6) racism | arBERT SE | XGB | 0.5765 | 0.7900 | arBERT SE | XGB | 0.5765 | 0.7900 |
| (6) xenophobia | mlBERT SE | LR | 0.5821 | 0.8700 | mlBERT SE | LR | 0.5821 | 0.8700 |
| (6) sexism | arBERT SE | XGB | 0.6678 | 0.8400 | arBERT SE | XGB | 0.6678 | 0.8400 |
| (6) religion | arBERT SE | XGB | 0.6094 | 0.9400 | n-grams | LR | 0.6464 | 0.8900 |
| (6) ableism | arBERT SE | RF | 0.4681 | 0.8800 | arBERT SE | RF | 0.4681 | 0.8800 |
| (6) classism | arBERT SE | RF | 0.4475 | 0.8100 | arBERT SE | RF | 0.4475 | 0.8100 |
| (6) ideologism | mlBERT SE | LR | 0.6484 | 0.6800 | mlBERT SE | LR | 0.6658 | 0.7000 |
| (6) other | n-grams | XGB | 0.6464 | 0.8900 | arBERT SE | XGB | 0.6094 | 0.9400 |

Table 10: Performance comparison of offensive language classification with and without sentiment analysis (SA) on the reFarad-500 dataset (best scores are marked in gray).

| level | repr. | model | F1 | acc | level | repr. | model | F1 | acc |
|---|---|---|---|---|---|---|---|---|---|
| (2) target | arBERT SE | XGB | 0.4783 | 0.7300 | (6) homophobia | arBERT SE | XGB | 0.6299 | 0.9500 |
| (3) target presence | n-grams | XGB | 0.6250 | 0.7600 | (6) sexism | arBERT SE | XGB | 0.6678 | 0.8400 |
| (4) vulgar | n-grams | LR | 0.6863 | 0.8400 | (6) other | arBERT SE | XGB | 0.6094 | 0.9400 |
| (5) hate speech/insult | arBERT SE | XGB | 0.7013 | 0.7300 | (6) ageism | arBERT SE | XGB | 1.0000 | 1.0000 |
| (5) discredit | ngrams | XGB | 0.7220 | 0.7300 | (6) ableism | arBERT SE | RF | 0.4681 | 0.8800 |
| (5) threat | arBERT SE | RF | 0.4975 | 0.9900 | (6) classism | arBERT SE | RF | 0.4475 | 0.8100 |
| (6) racism | arBERT SE | XGB | 0.5504 | 0.7900 | (6) ideologism | mlBERT SE | LR | 0.6349 | 0.6700 |
| (6) xenophobia | mlBERT SE | LR | 0.5821 | 0.8700 | (6) religion | n-grams | LR | 0.6464 | 0.8900 |

Table 11: Performance of offensive language classification with combined SA and ED enhancement on the reFarad-500 dataset.

## 5 Conclusions

This paper studies various levels of offensive language in Arabic following the ArSOL taxonomy of explicit offensive language. For this purpose, we re-annotate the existing dataset of (Liebeskind et al., 2024) to produce a quality dataset reFarad-500 covering multiple categories of Arabic offensive language. By applying various deep learning models, we assessed their effectiveness in detecting offensive content, and our experiments demonstrated that transformer models outperform traditional classifiers, highlighting their potential for this task. We also explored emotion detection and sentiment analysis to capture the emotional tone and subjective sentiment of offensive texts, showing that these methods are not merely auxiliary but integral to a comprehensive offensive language detection framework. The reFarad-500 dataset, together with full annotation guidelines, is freely available for research purposes at `https://github.com/NataliaVanetik/OffensiveLanguageDatasetInArabicFinegrainAnnotation`.

Future research should focus on expanding the dataset, integrating additional language resources, and enhancing classification models. The proposed taxonomy and annotation framework are designed to be adaptable, making them applicable not only to other Arabic dialects but also to languages with similar challenges in computational resources, data availability, and linguistic tooling. By combining taxonomy-driven annotation with semantic signals such as sentiment and emotion, this work offers a transferable foundation for offensive language detection across diverse linguistic contexts.

## Ethics Statement and Limitations

This study re-annotates publicly available Arabic offensive language data from online platforms such as Twitter and Facebook, using only anonymized texts without identifiable information. Native Arabic speakers, trained with comprehensive guidelines, conducted the re-annotation to ensure consistency, minimize bias, and maintain cultural sensitivity. We acknowledge the subjective nature of offense and encourage ethical consideration when using models trained on this dataset.

The reFarad-500 dataset has several limitations: its size (500 texts) restricts large-scale model training and category diversity; the focus on explicit offense excludes implicit cases for future work; category distributions remain imbalanced to preserve real-world patterns; and coverage is limited to Modern Standard Arabic and Levantine dialects, reducing generalizability to underrepresented varieties such as Maghrebi or Gulf Arabic.

## References


Ashraf Ahmad, Mohammad Azzeh, Eman Elnagi, Qasem Abu Al-Haija, Dana Halabi, Abdullah Aref, and Yousef AbuHour. 2024. Hate Speech Detection in the Arabic Language: Corpus Design, Construction and Evaluation. *Frontiers in Artificial Intelligence*, 7:1345445.

Azalden Alakrot, Liam Murray, and Nikola S Nikolov. 2018. Dataset construction for the detection of anti-social behaviour in online communication in Arabic. *Procedia Computer Science*, 142:174–181.

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the Arabic twittersphere. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 69–76. IEEE.

Ali Alhazmi. 2023. Hate Speech Dataset for the Saudi Dialect. Mendeley Data.

Amer J Almahdawi and William J Teahan. 2019. A new arabic dataset for emotion recognition. In *Intelligent Computing-Proceedings of the Computing Conference*, pages 200–216. Springer.

Maha Jarallah Althobaiti. 2022. BERT-based approach to Arabic hate speech and offensive language detection in Twitter: exploiting emojis and sentiment analysis. *International Journal of Advanced Computer Science and Applications*, 13(5).

Mohamed Aly and Amir Atiya. 2013. LABR: A large scale arabic book reviews dataset. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 494–498, Sofia, Bulgaria. Association for Computational Linguistics.

Amr Al-Khatib and Samhaa R. El-Beltagy. 2018. *Emotional Tone Detection in Arabic Tweets*, volume 10762 of *Lecture Notes in Computer Science*, page 105–114. Springer, Cham.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Flor Miriam Plaza-del Arco, Sercan Halat, Sebastian Padó, and Roman Klinger. 2021. Multi-task learning with sentiment, emotion, and target detection to recognize hate speech and offensive language. *arXiv preprint arXiv:2109.10255*.

Abdullah Aref, Rana Husni Al Mahmoud, Khaled Taha, Mahmoud Al-Sharif, et al. 2020. Hate speech detection of arabic shorttext. In *9th International Conference on Information Technology Convergence and Services (ITCSE 2020)*, pages 81–94. Computer Science & Information Technology.

Tianqi Chen. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4.

Shammur Absar Chowdhury, Hamdy Mubarak, Ahmed Abdelali, Soon-gyo Jung, Bernard J Jansen, and Joni Salminen. 2020. A multi-platform Arabic news comment dataset for offensive language detection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6203–6212.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

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. Association for Computational Linguistics.

Abdelrahim Elmadany, Chiyu Zhang, Muhammad Abdul-Mageed, and Azadeh Hashemi. 2020. Leveraging affective bidirectional transformers for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 102–108.

Kabil Essefar, Hassan Ait Baha, Abdelkader El Mahdaouy, Abdellah El Mekki, and Ismail Berrada. 2023. OMCD: Offensive Moroccan Comments Dataset. *Language Resources and Evaluation*, pages 1–21.

Hatem Haddad, Hala Mulki, and Asma Oueslati. 2019. T-HSAB: A Tunisian hate speech and abusive dataset. In *International Conference on Arabic Language Processing*, pages 251–263. Springer.

Go Inoue, Hassan Sajjad, Fath Elrahman Saleh, Abdelrahim Elmadany, and Preslav Nakov. 2021. Camelbert: Open large-scale language models for arabic. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4706–4721. Association for Computational Linguistics.

Kirolos Khela. 2023. Sentiment-bert: Arabic sentiment analysis model. `https://huggingface.co/kiroloskhela/Sentiment-Bert`. Accessed: 2025-07-21.

David G Kleinbaum, K Dietz, M Gail, Mitchel Klein, and Mitchell Klein. 2002. *Logistic regression*. Springer.

Barbara Lewandowska-Tomaszczyk. 2023. A simplified taxonomy of offensive language (SOL) for computational applications. *Konin Language Studies*, 10(3):213–227.

Barbara Lewandowska-Tomaszczyk, Anna Bączkowska, Chaya Liebeskind, Giedre Valunaite Oleskeviciene, and Slavko Žitnik. 2023. An integrated explicit and implicit offensive language taxonomy. *Lodz Papers in Pragmatics*, 19(1):7–48.

Barbara Lewandowska-Tomaszczyk, Slavko Žitnik, Anna Bączkowska, Chaya Liebeskind, Jelena Mitrović, and Giedrė Valūnaitė Oleškevičienė. 2021. LOD-connected offensive language ontology and tagset enrichment. In *R. Carvalho & R. Rocha Souza, R. (Eds.), Proceedings of the Workshop and Tutorials held at LDK 2021 co-located with the 3rd Language, Data and Knowledge Conference*, volume 3064, pages 135–150. CEUR Workshop Proceedings.

Barbara Lewandowska-Tomaszczyk, Slavko Žitnik, Chaya Liebeskind, Giedrė Valūnaitė-Oleškevičienė, Anna Bączkowska, Paul A Wilson, Marcin Trojszczak, Ivana Brač, Lobel Filipić, and Ana Ostroški Anić. 2022. Annotation scheme and evaluation: The case of offensive language. *Rasprave*.

Chaya Liebeskind, Ali Afawi, Marina Litvak, and Natalia Vanetik. 2024. Classifying offensive language in arabic: a novel taxonomy and dataset. *Lodz Papers in Pragmatics*, 20(2):433–462.

Chaya Liebeskind, Natalia Vanetik, and Marina Litvak. 2023. Hebrew offensive language taxonomy and dataset. *Lodz Papers in Pragmatics*, 19(2):325–351.

Marina Litvak, Natalia Vanetik, Yaser Nimer, Abdulrhman Skout, and Israel Beer-Sheba. 2021. Offensive language detection in semitic languages. In *Multimodal Hate Speech Workshop*, volume 2021, pages 7–12.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2023a. Hate speech and offensive language detection using an emotion-aware shared encoder. In *ICC 2023-IEEE International Conference on Communications*, pages 2852–2857. IEEE.

Khouloud Mnassri, Praboda Rajapaksha, Reza Farahbakhsh, and Noël Crespi. 2023b. Hate speech and offensive language detection using an emotion-aware shared encoder. *CoRR*, abs/2302.08777.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on Arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Hamdy Mubarak, Ammar Rashed, Kareem Darwish, Younes Samih, and Ahmed Abdelali. 2020. Arabic offensive language on Twitter: Analysis and experiments. *arXiv preprint arXiv:2004.02192*.

Hala Mulki and Bilal Ghanem. 2021. Let-Mi: An Arabic Levantine Twitter Dataset for Misogynistic Language. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 154–163.

Hala Mulki, Hatem Haddad, Chedi Bechikh Ali, and Halima Alshabani. 2019. L-hsab: A Levantine Twitter dataset for hate speech and abusive language. In *Proceedings of the third workshop on abusive language online*, pages 111–118.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. ASTD: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal. Association for Computational Linguistics.

Hatem Noaman. 2023. Improved emotion detection framework for arabic text using transformer models. *Advanced Engineering Technology and Application*, 12(2):1–11.

Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. 2019. Multilingual and multi-aspect hate speech analysis. 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 4675–4684.

Mahesh Pal. 2005. Random forest classifier for remote sensing classification. *Journal of Remote Sensing*, 26(1):217–222.

Niloofar Safi Samghabadi, Afsheen Hatami, Mahsa Shafaei, Sudipta Kar, and Thamar Solorio. 2020. Attending the emotions to detect online abusive language. In *Proceedings of the Fourth Workshop on Online Abuse and Harms*, pages 79–88.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts. *arXiv*.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. 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 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Cagri Coltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447. Association for Computational Linguistics.

# Measuring Prosodic Richness in LLM-Generated Responses for Conversational Recommendation

**Darshna Parmar, Pramit Mazumdar**
Department of Computer Science and Engineering
Indian Institute of Information Technology Vadodara
{darshna.parmar, pramit.mazumdar}@iiitvadodara.ac.in

## Abstract

This paper presents a novel framework for stylistic evaluation in conversational recommendation systems (CRS), focusing on the prosodic and expressive qualities of generated responses. While prior work has predominantly emphasized semantic relevance and recommendation accuracy, the stylistic fidelity of model outputs remains underexplored. We introduce the prosodic richness score (PRS), a composite metric that quantifies expressive variation through structural pauses, emphatic lexical usage, and rhythmic variability. Using PRS, we conduct both sentence-level and turn-level analyses across six contemporary large language models (LLMs) on two benchmark CRS datasets: ReDial, representing goal-oriented dialogue, and INSPIRED, which incorporates stylized social interaction. Empirical results reveal statistically significant differences ($p < 0.01$) in PRS between human and model-generated responses, highlighting the limitations of current LLMs in reproducing natural prosodic variation. Our findings advocate for broader evaluation of stylistic attributes in dialogue generation, offering a scalable approach to enhance expressive language modeling in CRS.

## 1 Introduction

Conversational Recommendation Systems (CRS) aim to provide personalized recommendations through natural language dialogue (Jannach et al., 2021). With the emergence of large language models (LLMs), such as LLaMA, Mistral, and Gemini, the quality of generated dialogue has significantly improved in terms of coherence, informativeness, and contextual relevance (Thoppilan et al., 2022; Li et al., 2018; Numaya et al., 2025). However, most existing CRS evaluation methods emphasize automatic metrics—such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), recommendation accuracy, and human evaluation (Liu et al., 2020)—while neglecting stylistic features, particularly prosodic aspects like rhythm, expressiveness, and tone.

Prosody—referring to rhythm, emphasis, and structural variation in language—is a key component of natural human communication (Ladd and Arvaniti, 2023). In text-based dialogue systems, prosodic features manifest through punctuation (e.g., pauses via commas or periods), lexical emphasis (e.g., adjectives, adverbs, interjections), sentence rhythm (e.g., variance in sentence length), and expressive markers (e.g., questions and exclamations). These cues contribute to the emotional resonance and naturalness of responses, making them especially important for CRS applications aiming to simulate engaging human-like dialogue or support voice interfaces.

Despite recent advances, current LLM-based systems often generate stylistically flat responses lacking variation in tone, structure, or emphasis—limiting engagement and diminishing perceived human-likeness. Existing evaluation frameworks fail to capture these nuanced yet important dimensions of expressiveness (Li et al., 2024; Park et al., 2024). To address this gap, we propose a novel method for analyzing the prosodic quality of LLM-generated responses in CRS. Specifically, we quantify expressiveness using three interpretable features: (i) structural pauses (e.g., punctuation frequency), (ii) lexical emphasis (e.g., adjectives, adverbs, interjections), and (iii) rhythm variation (variance in sentence lengths). PRS provides a normalized score that captures the richness of a response's structure and delivery style. To the best of our knowledge, this is the first systematic attempt to assess textual prosody in CRS responses.

We investigate the stylistic expressiveness of dialogue responses generated by six recent LLMs within a standard CRS framework. Our evaluation focuses on models such as LLaMA, Gemma,



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 120–130, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-014

Gemini, Qwen, and Mistral, assessing their ability to produce responses with natural prosodic qualities. The CRS model is trained and evaluated on the standard splits of two benchmark datasets: ReDial (Li et al., 2018), which is goal-oriented and centers on movie recommendations, and INSPIRED (Hayati et al., 2020), which emphasizes emotionally rich conversations. We analyze the LLM-generated responses using both sentence-level and turn-level PRS distributions, compare them with human-authored ground-truths, and examine stylistic variation across dialogue turns.

This enables a more holistic evaluation of LLMs in CRS by capturing stylistic expressiveness often overlooked by standard automatic and human evaluation methods.

Our key contributions are as follows:

- We introduce the Prosodic Richness Score (PRS), a novel metric designed to quantify stylistic and prosodic expressiveness in dialogue responses.
- We evaluate PRS on responses generated by six state-of-the-art LLMs, using a standard CRS framework applied to two benchmark datasets: ReDial and INSPIRED.
- Our analysis uncovers consistent prosodic gaps between LLM-generated and human-authored responses, highlighting the limitations of current models in producing naturally expressive dialogue.

This work offers a new lens for evaluating conversational agents, emphasizing not only what is generated but how it is said. Our findings underscore the need for more prosody-aware generation techniques to bridge the gap between human and machine dialogue.

## 2 Related Work

While LLMs have enhanced CRSs in terms of relevance and coherence, prior work has largely emphasized task-oriented metrics like recommendation accuracy. In contrast, stylistic and prosodic factors—crucial for naturalness and user engagement—remain underexplored. This section reviews related research on CRS evaluation, stylistic and affective language generation, and the role of prosody in natural language processing.

### 2.1 Evaluation of Conversational Recommender Systems

CRSs aim to provide personalized suggestions through dialogue (Christakopoulou et al., 2016; Li et al., 2018; Jannach et al., 2021). Traditional evaluation relies on task-specific metrics like BLEU, ROUGE, or Recall@K, which overlook stylistic richness and perceived naturalness. While human evaluations and learned reward models offer deeper insights (See et al., 2019; Ghazarian et al., 2022), they are costly and less interpretable. Recent reference-free evaluators, such as FACE (Chen et al., 2025), show strong alignment with human judgments, and others demonstrate robustness under adversarial settings (Vasselli et al., 2025). In contrast, our PRS provides a lightweight and interpretable measure of stylistic expressiveness.

### 2.2 Stylistic and Affective Generation in Dialogue

Stylistic elements such as tone, emotion, and personality play a vital role in enhancing conversational engagement(Qian et al., 2023; Ma et al., 2024). Dataset like Empathetic Dialogues (Rashkin et al., 2019) has highlighted the importance of generating affective and stylistically rich responses, particularly in CRS. Prior approaches to stylistic control in text generation have leveraged lexical constraints (Iso, 2024), persona-based latent attribute control (Lu et al., 2023), and decoding-time mechanisms via dynamic attribute graphs (Liang et al., 2024). However, evaluation metrics for stylistic expressiveness remain limited. We address this gap by introducing PRS, which facilitates cross-LLM comparisons of stylistic variation in CRS outputs, with a focus on the ReDial and INSPIRED dataset.

### 2.3 Prosody in Text and Speech Systems

Prosody—encompassing rhythm, emphasis, and structural variation—plays a crucial role in human communication and is extensively studied in speech synthesis(Li et al., 2025; Raitio et al., 2022). While TTS models incorporate prosodic control(Liu et al., 2024; Que and Ragni, 2025), textual dialogue evaluation rarely considers prosodic features. Some studies utilize prosodic cues for emotion recognition or discourse segmentation(Wei et al., 2023; Prévot and Wang, 2024), but these often rely on acoustic inputs. Our work is distinct in applying prosody-based analysis directly to text, enabling stylistic evaluation of LLM-generated responses in CRS.

While previous work has made strides in affective dialogue generation and prosody modeling—particularly in speech applications—there remains a lack of principled, text-based analysis for quantifying prosodic expressiveness in generated responses. Our work bridges this gap by introducing a simple yet effective measure, the PRS, enabling scalable, interpretable evaluation of stylistic quality in LLM-generated responses across diverse CRS datasets.

## 3 Methodology

This section outlines our approach to evaluating the stylistic and prosodic expressiveness of responses generated by LLMs in conversational recommendation systems. We introduce a lightweight linguistic framework that extracts text-based prosodic features and computes a unified PRS to assess variation and naturalness across both human-authored ground-truths and model-generated responses.

### 3.1 Datasets

We use two publicly available CRS datasets. ReDial (Li et al., 2018) contains over 10,000 goal-driven movie recommendation dialogues with annotated movie mentions, emphasizing task performance. In contrast, INSPIRED (Hayati et al., 2020) includes 1,001 open-domain dialogues enriched with tone annotations (e.g., empathetic, humorous), supporting evaluation of affective and stylistic expressiveness. Together, these datasets enable both functional and stylistic analysis.

### 3.2 LLM-Generated Response Collection

To analyze stylistic variation in generated dialogue, we evaluate responses from six large language models: llama-3.1-8b-instant (Touvron et al., 2023), llama-3.2-3b-preview (Touvron et al., 2023), gemma2-9b-it (Anil et al., 2024), gemini-1.5-flash-8b (Google DeepMind, 2024), qwen-2.5-32b (Inc., 2024), and mistral-saba-24b (Jiang et al., 2024). The selected models span a broad spectrum of capacities, ranging from 3B to 32B parameters, and include both decoder-only and multimodal architectures. This range ensures coverage of both lightweight and high-capacity LLMs, allowing us to examine whether expressive richness in generated responses is consistently maintained across models of varying size and complexity. Each model is prompted to produce a single-turn response given a user utterance from either the ReDial or INSPIRED dataset. Each generated output is aligned with its corresponding human-authored ground-truth response, enabling direct comparison of prosodic and stylistic characteristics. For transparency and reproducibility, we include the exact prompts used during LLM-based response generation in Appendix A.

### 3.3 Text-Based Prosody Feature Extraction

To quantify the stylistic expressiveness, we compare responses generated by LLMs against human-written ground-truth responses from established CRS datasets. We extract a set of interpretable textual features that serve as proxies for prosodic expressiveness in dialogues generated via various LLMs. Specifically, we compute:

- Pause count: The number of punctuation-based structural markers (e.g., periods, commas, semicolons, question marks, and exclamation marks), which simulate natural pauses in spoken language.

- Emphasis count: The number of expressive lexical items—identified via part-of-speech tags such as adjectives, adverbs, and interjections—that often signal emotional tone or subjective emphasis.

- Rhythm variance: The variance in sentence length (in tokens), reflecting diversity in syntactic structure and rhythmic flow.

- Question and exclamation counts: The number of interrogative and exclamatory sentences, capturing tone variability and conversational dynamism.

- Sentence count: The number of distinct sentences in a response, offering a basic structural measure of length and complexity.

While PRS captures stylistic variation through pause, emphasis, and rhythm features, we acknowledge that overuse of these markers could artificially inflate scores. All features are extracted using a spaCy-based linguistic preprocessing pipeline, including sentence segmentation and part-of-speech tagging, which ensures consistent and linguistically informed analysis across both model-generated and human-authored responses.

#### 3.3.1 Defining the Prosodic Richness Score

We define the PRS as a composite metric to capture the stylistic richness of a LLM generated response:

$$\mathrm{PRS} = \frac{1}{10}\,(0.4 \cdot \text{pause} + 0.3 \cdot \text{emph} + 0.3 \cdot \text{var}) \tag{1}$$

The score is normalized between 0 and 1 to support direct comparisons across models and datasets. A higher score reflects greater stylistic diversity and perceived naturalness in generated responses. The weights in Equation 1 are assigned based on empirical observations of the contribution of each feature to stylistic expressiveness, with pause count receiving slightly greater weight due to its consistent role in conveying natural rhythm.

### 3.4 Prosody-Aware Evaluation Method

We propose a multi-level evaluation framework based on the PRS to assess the stylistic quality of LLM-generated dialogue. At the sentence level, PRS captures local prosodic expressiveness by comparing model outputs to human references. At the turn level, we analyze PRS evolution across dialogue turns to identify stylistic consistency or degradation over time. For model-wise comparison, we compute the average PRS for each LLM and benchmark it against human-authored baselines. To evaluate the statistical significance of stylistic differences, we conduct paired $t$-tests on the PRS distributions of model and human responses.

The final PRS is computed as defined in Equation 1. This score is calculated for both model-generated utterances (*model_PRS*) and the human-written ground truth responses (*gt_PRS*) at the sentence level. Although the reference sentences are drawn from the original ReDial and INSPIRED datasets—resources that may have been partially seen during pretraining—they serve as domain-aligned and affect-rich baselines for stylistic evaluation. This prosody-aware framework thus enables a linguistically grounded, fine-grained, and interpretable assessment of expressiveness in conversational recommendation systems.

## 4 Results & Discussion

To assess the stylistic expressiveness of LLMs in CRSs, we analyze PRS at both the sentence and turn levels. Our experiments aim to answer the following research questions:

- **RQ1:** How do different LLMs compare to human responses in terms of sentence-level stylistic expressiveness?
- **RQ2:** How does the prosodic richness of model-generated responses vary across successive dialogue turns?
- **RQ3:** To what extent do LLMs sustain stylistic diversity throughout an interaction compared to human-written dialogues?
- **RQ4:** What are the model-specific trends in stylistic degradation or consistency, and which models demonstrate stronger ability to retain prosodic richness?
- **RQ5:** Which LLM demonstrates the highest overall stylistic richness?

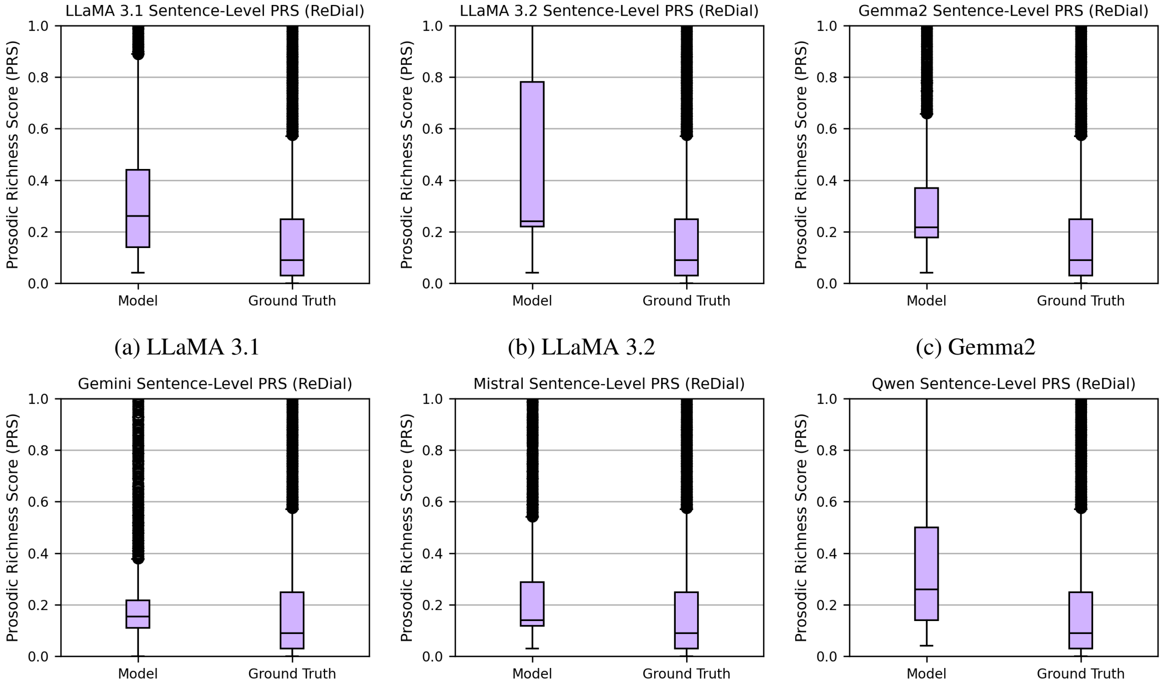
These questions guide our analysis of the expressive capacity of LLMs using PRS as a stylistic evaluation metric. We use both sentence-level and turn-level granularity to investigate how well models emulate human-like variation in tone, rhythm, and emphasis across CRS interactions.

### 4.1 RQ1: Sentence-Level Stylistic Expressiveness

To capture fine-grained expressiveness in CRS outputs, we compute PRS at the sentence level. Sentence-level analysis is essential, as individual utterances shape tone, emotional resonance, and user engagement—particularly in affect-rich dialogues. This granularity helps assess how well LLMs emulate human-like prosodic variation.

Figures 1 and 2 show PRS distributions across six LLMs. In both datasets, human responses consistently exhibit greater stylistic richness, with higher medians and broader variance. The gap is especially pronounced in INSPIRED, which contains emotionally expressive, tone-sensitive dialogue.

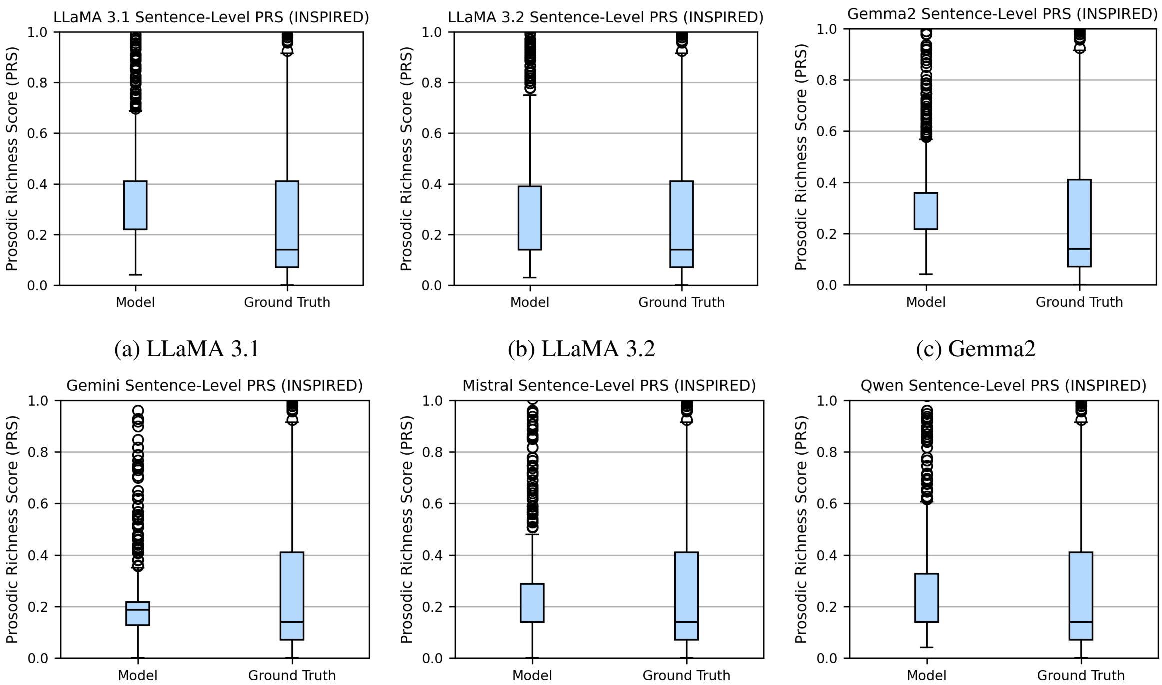
Among the models, LLaMA 3.1 and 3.2 display relatively higher prosodic variation, while Gemini, Mistral, and Qwen lag behind. Model outputs also show fewer outliers, highlighting their limited expressive variability compared to human responses. These results suggest that while LLMs produce contextually relevant outputs, they often lack the stylistic nuance found in human dialogue. Sentence-level PRS thus provides a valuable diagnostic for evaluating expressive quality and highlights the need for better stylistic modeling in CRS systems.




(a) LLaMA 3.1 (b) LLaMA 3.2 (c) Gemma2

(d) Gemini (e) Mistral (f) Qwen

Figure 1: Sentence-level PRS distributions for ReDial (RQ1), comparing prosodic variation in LLM-generated responses and human-authored references.




(a) LLaMA 3.1 (b) LLaMA 3.2 (c) Gemma2

(d) Gemini (e) Mistral (f) Qwen

Figure 2: Sentence-level PRS distributions for INSPIRED (RQ1), comparing prosodic variation in LLM-generated responses and human-authored references.

### 4.2 RQ2: PRS Across Dialogue Turns

To assess how stylistic expressiveness evolves throughout a conversation, we compute PRS at the turn level. Unlike sentence-level analysis, turn-level PRS captures the aggregate expressiveness of all utterances within a dialogue turn, revealing how consistently an LLM maintains stylistic richness across interactions. Turn-level PRS is computed by averaging sentence-level PRS scores within each turn, yielding two PRS sequences per conversation—one for *model_PRS* and one for *gt_PRS*. These plots capture the progression of prosodic richness throughout the conversation.

Figures 3 and 4 illustrate a consistent decline in turn-level PRS for most LLMs, indicating diminished stylistic expressiveness over extended interactions. In contrast, human responses exhibit greater stability and variation, with the model-human gap widening in later turns. Among models, LLaMA 3.2 demonstrates stronger stylistic consistency than Gemini or Mistral, reflecting architectural and training differences. These results underscore the value of turn-level PRS in capturing temporal expressiveness—an essential dimension for affect-rich conversational systems—and motivate the development of style-aware, turn-sensitive generation strategies.

*Case-wise Illustration:* Appendix B present turn-level PRS patterns for single conversations from ReDial (ConvID: 22709) and INSPIRED (ConvID: 20191127-224739_530_live.pkl alise as 001), respectively. In both cases, LLaMA 3.2 and Gemma2 closely follow human PRS trends, while Gemini and Mistral display flatter or inconsistent profiles, reflecting reduced ability to sustain stylistic expressiveness across turns.

### 4.3 RQ3–RQ5: Model-Specific Stylistic Trends

To investigate the consistency of stylistic expression across dialogue progression (RQ3), patterns of degradation or stability (RQ4), and overall prosodic richness across models (RQ5), we conduct a comprehensive analysis of sentence- and turn-level PRS across six state-of-the-art LLMs using the ReDial and INSPIRED datasets.

Figure 5 presents a unified visualization of turn-wise evolution and model-level PRS gaps. The turn-wise plots reveal that human-authored responses exhibit consistently higher and more stable prosodic richness throughout the conversation. In contrast, most LLMs show a gradual decline in PRS, particularly in later turns, indicating a loss of stylistic variation over time. Notably, LLaMA 3.2 and Gemma2 demonstrate comparatively stronger stylistic consistency, while Gemini and Mistral show flatter or declining trends, reflecting limited ability to preserve expressive variation.

The model-wise comparison of aggregate PRS values relative to human ground truth. Across both datasets, even the most competitive models fall short of human-level prosodic expressiveness, underscoring persistent limitations in current LLMs. These findings highlight the need for prosody-aware generation strategies that explicitly model expressive diversity and sustain stylistic richness across dialogue turns.

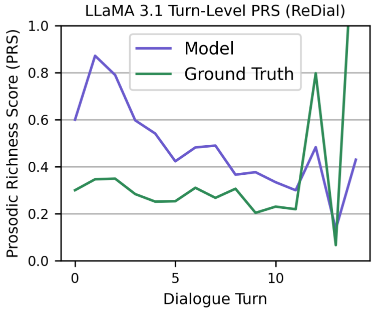
These results, consistent across both datasets, suggest that while LLMs achieve surface-level fluency, they often under utilize stylistic features such as lexical emphasis and rhythm variation, particularly in emotionally expressive contexts like INSPIRED.

## 5 Statistical Validation of Prosodic Richness Differences

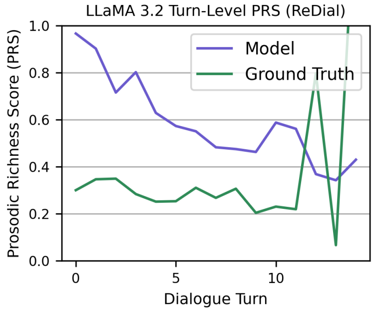
To assess whether LLM-generated responses differ significantly in stylistic expressiveness compared to human-authored dialogue, we conducted a paired $t$-test on sentence-level PRS. For each model, PRS values were paired with corresponding ground-truth responses across both ReDial and INSPIRED datasets.

The results, summarized in Table 1, reveal several key findings:

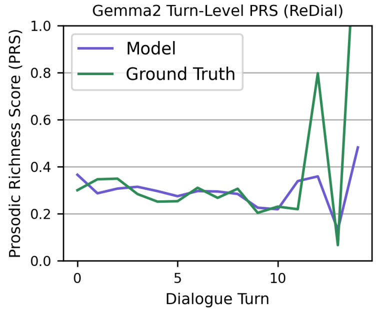
- Most models (e.g., LLaMA 3.1, LLaMA 3.2, Gemini, Qwen) show statistically significant differences ($p < 0.05$) from human responses on ReDial, indicating consistent stylistic divergence.
- On the INSPIRED dataset, however, fewer models show significance, suggesting either reduced expressiveness in the models or more variability in human references.
- LLaMA 3.2 shows significant differences across both datasets, indicating high stylistic deviation despite its strong median PRS.
- In contrast, models like Gemma 2 and Qwen do not differ significantly on INSPIRED, implying closer alignment to human style or reduced variance in outputs.
- Gemini consistently shows negative $t$-values (e.g., $t = -4.54$, $p < 0.0001$ on ReDial), indicating it underperforms in stylistic richness relative to human-written dialogue.



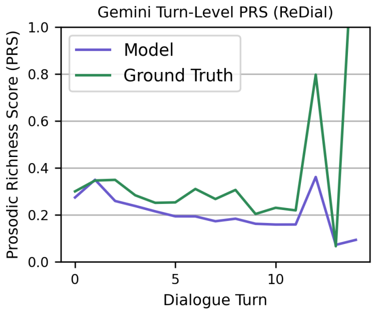

(a) LLaMA 3.1



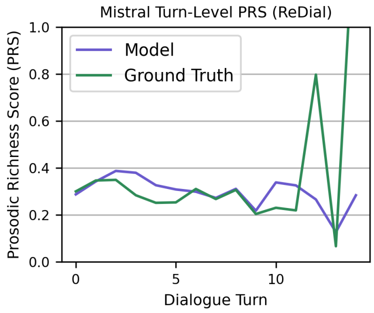

(b) LLaMA 3.2



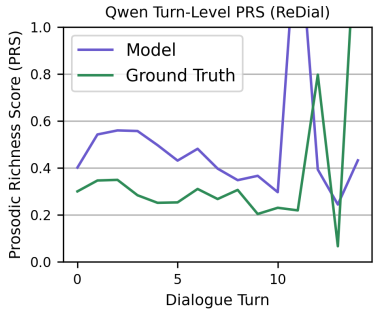

(c) Gemma2




(d) Gemini

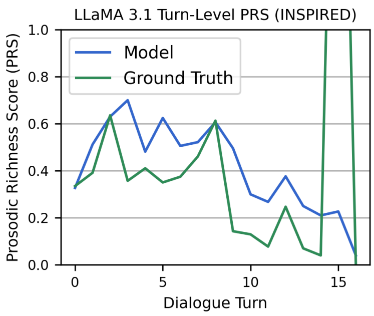



(e) Mistral

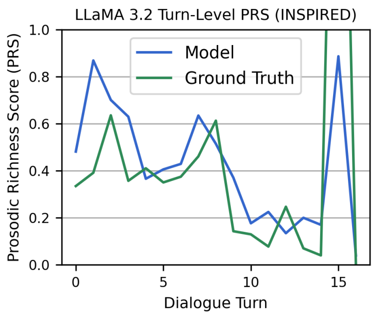



(f) Qwen

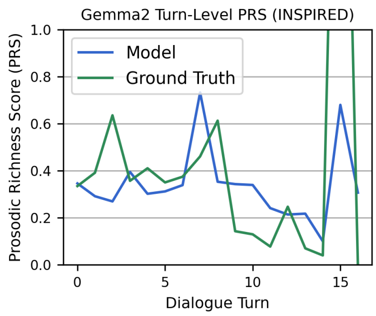
Figure 3: Turn-level PRS comparisons for ReDial (RQ2), showing how stylistic expressiveness progresses across dialogue turns in LLM-generated and human-authored responses.



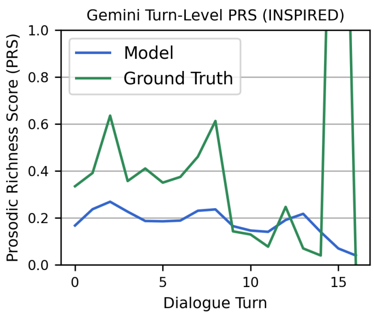

(a) LLaMA 3.1



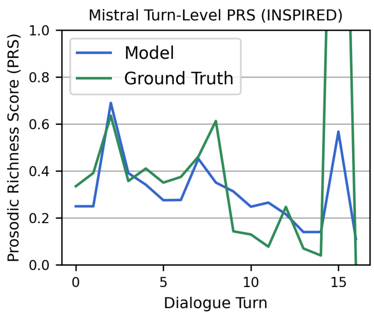

(b) LLaMA 3.2



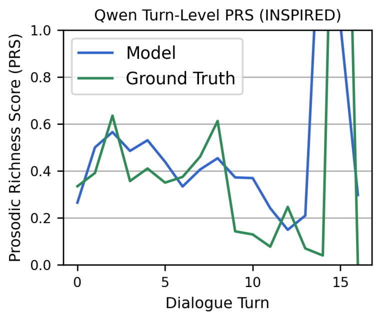

(c) Gemma2




(d) Gemini



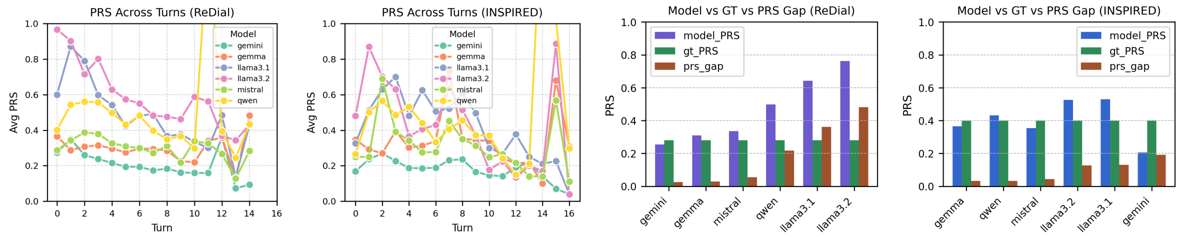

(e) Mistral




(f) Qwen

Figure 4: Turn-level PRS comparisons for INSPIRED (RQ2), showing how stylistic expressiveness progresses across dialogue turns in LLM-generated and human-authored responses.

These findings statistically validate that while some LLMs approach human-level prosody, a measurable and significant expressiveness gap still exists. This supports the use of PRS as a diagnostic tool and highlights the need for prosody-aware training methods.




(a) (b) (c) (d)

Figure 5: Visualizations addressing RQ3–RQ5 over ReDial and INSPIRED datasets. (a–b) presents turn-level PRS across dialogue turns, capturing patterns of stylistic consistency and degradation (RQ3, RQ4). (c–d) presents aggregate model-wise PRS comparisons relative to human-authored ground truth, highlighting persistent gaps in expressive richness across LLMs (RQ5).

Table 1: Paired $t$-test results comparing sentence-level PRS between LLM-generated and human responses across ReDial and INSPIRED datasets.

| **Model** | **ReDial** | | | **INSPIRED** | | |
|---|---|---|---|---|---|---|
| | **t-value** | **p-value** | **Significance** | **t-value** | **p-value** | **Significance** |
| LLaMA 3.1 | 11.4 | <0.0001 | Yes | 3.13 | <0.0018 | Yes |
| LLaMA 3.2 | 23.96 | <0.0001 | Yes | 2.22 | <0.0268 | Yes |
| Gemma 2 | 1.19 | <0.2321 | No | -0.72 | <0.4713 | Yes |
| Gemini | -4.54 | <0.0001 | Yes | -6.2 | <0.0001 | Yes |
| Mistral | 3.84 | <0.0001 | Yes | -0.91 | <0.3611 | No |
| Qwen | 17.09 | <0.0001 | Yes | 0.83 | <0.4078 | No |

## 6 Conclusion

We introduced PRS as a stylistic metric to assess LLM-generated responses in conversational recommendation systems. Through sentence- and turn-level analysis on the ReDial and INSPIRED datasets, we found that while LLMs produce coherent responses, they often lack the stylistic richness and variation characteristic of human dialogue—particularly in extended interactions. Among the evaluated models, LLaMA 3.2 demonstrated the highest prosodic expressiveness, whereas Gemini and Mistral lagged behind. These findings underscore the importance of integrating prosodic and stylistic diversity into future CRS models to enable more engaging and human-like conversations.

## Limitations

Our study evaluates prosodic expressiveness in CRS responses using PRS, but PRS is not integrated into generation or used for stylistic enhancement. No LLM is fine-tuned with PRS supervision, limiting its direct impact. While benchmarked against human responses, PRS is not yet validated with independent human ratings, and it relies on surface-level textual proxies. Incorporating higher-level prosodic cues or spoken responses could provide a more robust assessment of expressiveness in future work.

## Ethics Statement

This work involves the analysis of model-generated and human-written conversational data using publicly available datasets: ReDial and INSPIRED. Both datasets are anonymized and curated for research use, and no personally identifiable information (PII) is processed. Our methodology does not involve human subjects or new data collection. However, we acknowledge that automatic evaluation of stylistic expressiveness may carry inherent biases based on dataset demographics and model training corpora. We urge caution in deploying stylistically expressive models in sensitive domains such as mental health or education, where unintended emotional tone may have real-world consequences.

## References


Rohan Anil, Yiding Jiang, et al. 2024. Gemma: Lightweight, state-of-the-art open models. https://ai.google.dev/gemma.

Nuo Chen, Quanyu Dai, Xiaoyu Dong, Xiao-Ming Wu, and Zhenhua Dong. 2025. Face: A fine-grained reference free evaluator for conversational recommender systems. *arXiv preprint arXiv:2501.09493*.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824.

Sarik Ghazarian, Behnam Hedayatnia, Alexandros Papangelis, Yang Liu, and Dilek Hakkani-Tur. 2022. What is wrong with you?: Leveraging user sentiment for automatic dialog evaluation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4194–4204, Dublin, Ireland. Association for Computational Linguistics.

Google DeepMind. 2024. Gemini 1.5 technical report. arXiv preprint arXiv:2403.05530.

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. 2020. Inspired: Toward sociable recommendation dialog systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8142–8152. Association for Computational Linguistics.

Baidu Inc. 2024. Qwen2: The next-gen language model family. https://github.com/QwenLM/Qwen.

Hayate Iso. 2024. AutoTemplate: A simple recipe for lexically constrained text generation. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 1–12, Tokyo, Japan. Association for Computational Linguistics.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. 2021. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):1–36.

Yujia Jiang, Guillaume Lample, et al. 2024. Mistral 7b. https://mistral.ai/news/mistral-7b/.

D Robert Ladd and Amalia Arvaniti. 2023. Prosodic prominence across languages. *Annual Review of Linguistics*, 9(1):171–193.

Jinpeng Li, Zekai Zhang, Quan Tu, Xin Cheng, Dongyan Zhao, and Rui Yan. 2024. Stylechat: Learning recitation-augmented memory in llms for stylized dialogue generation. *arXiv preprint arXiv:2403.11439*.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yinghao Aaron Li, Cong Han, and Nima Mesgarani. 2025. Styletts: A style-based generative model for natural and diverse text-to-speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 19(1):283–296.

Xun Liang, Hanyu Wang, Shichao Song, Mengting Hu, Xunzhi Wang, Zhiyu Li, Feiyu Xiong, and Bo Tang. 2024. Controlled text generation for large language model with dynamic attribute graphs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5797–5814, Bangkok, Thailand. Association for Computational Linguistics.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiaxuan Liu, Zhaoci Liu, Yajun Hu, Yingying Gao, Shilei Zhang, and Zhenhua Ling. 2024. Diffstyletts: Diffusion-based hierarchical prosody modeling for text-to-speech with diverse and controllable styles. *arXiv preprint arXiv:2412.03388*.

Zeming Liu, Haifeng Wang, Zheng-Yu Niu, Hua Wu, Wanxiang Che, and Ting Liu. 2020. Towards conversational recommendation over multi-type dialogs. *arXiv preprint arXiv:2005.03954*.

Zhenyi Lu, Wei Wei, Xiaoye Qu, Xian-Ling Mao, Dangyang Chen, and Jixiong Chen. 2023. Miracle: Towards personalized dialogue generation with latent-space multiple personal attribute control. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5933–5957, Singapore. Association for Computational Linguistics.

Zhiqiang Ma, Wenchao Jia, Yutong Zhou, Biqi Xu, Zhiqiang Liu, and Zhuoyi Wu. 2024. Personality enhanced emotion generation modeling for dialogue systems. *Cognitive Computation*, 16(1):293–304.

Ikumi Numaya, Shoji Moriya, Shiki Sato, Reina Akama, and Jun Suzuki. 2025. How stylistic similarity shapes preferences in dialogue dataset with user and third party evaluations. *arXiv preprint arXiv:2507.10918*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

ChaeHun Park, Minseok Choi, Dohyun Lee, and Jaegul Choo. 2024. Paireval: Open-domain dialogue evaluation with pairwise comparison. *arXiv preprint arXiv:2404.01015*.

Laurent Prévot and Sheng-Fu Wang. 2024. Investigating discourse segmentation in taiwan southern min spontaneous speech. In *5th Workshop on Computational Approaches to Discourse*, page 50.

Yushan Qian, Bo Wang, Shangzhao Ma, Wu Bin, Shuo Zhang, Dongming Zhao, Kun Huang, and Yuexian Hou. 2023. Think twice: A human-like two-stage conversational agent for emotional response generation. *arXiv preprint arXiv:2301.04907*.

Shumin Que and Anton Ragni. 2025. Visualspeech: Enhance prosody with visual context in tts. *arXiv preprint arXiv:2501.19258*.

Tuomo Raitio, Jiangchuan Li, and Shreyas Seshadri. 2022. Hierarchical prosody modeling and control in non-autoregressive parallel neural tts. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7587–7591.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. 2019. What makes a good conversation? how controllable attributes affect human judgments. 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 1702–1723, Minneapolis, Minnesota. Association for Computational Linguistics.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Y-Lan Boureau, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

Justin Vasselli, Adam Nohejl, and Taro Watanabe. 2025. Measuring the robustness of reference-free dialogue evaluation systems. *arXiv preprint arXiv:2501.06728*.

Xianhao Wei, Jia Jia, Xiang Li, Zhiyong Wu, and Ziyi Wang. 2023. A discourse-level multi-scale prosodic model for fine-grained emotion analysis. *arXiv preprint arXiv:2309.11849*.

# Appendix

## A Prompts Used for LLM Response Generation

### Case 1: No Recommendation Available

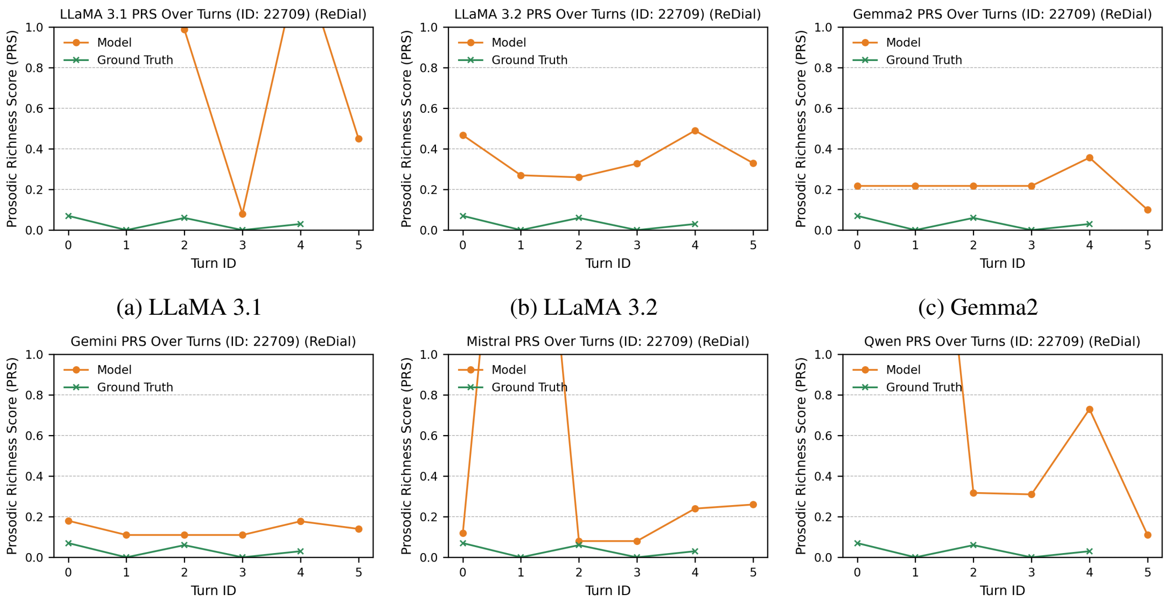
I will provide you with a user input that contains some sort of chit-chat or question. I want you to generate an output text that incorporates a sort of chit chat and then followed by some question related to movies, actors, genres etc.
**Example 1:** User Input: "Hi, how are you?" Output: "Hi! I'm doing well. What kind of movies are you looking for?"
Now, do a similar task for the given user input.

### Case 2: Recommendation Available

I will provide you with a user input that contains some movie names, actor names, cast, directors, genre, etc. Additionally, I will provide you with a recommendation that is relevant to the input. I want you to generate an output text that incorporates both the information from the user input and the recommendation.
**Example 1:** User Input: "I really liked Avengers and SpiderMan. They are both Thrillers and Tom Holland featured in both of them. Released in 2012 directed by Tarantino." Related Attributes: "Thor, Chris Hemsworth." Output: "You can watch Thor. It stars Chris Hemsworth and is similar to the Avengers."
**Example 2:** If user recommendation is empty then ask the user a relevant question about their likings regarding genres, casts etc and engage with the user.
**Example 3:** If the user input is present and some ambiguity is present regarding the recommendation generated then clarify it with the user by asking more specific questions regarding the cast, year of release etc. Now, do a similar task for the given user input and recommendation.

## B Case-wise Turn-Level PRS Analysis over ReDial and INSPIRED Datasets

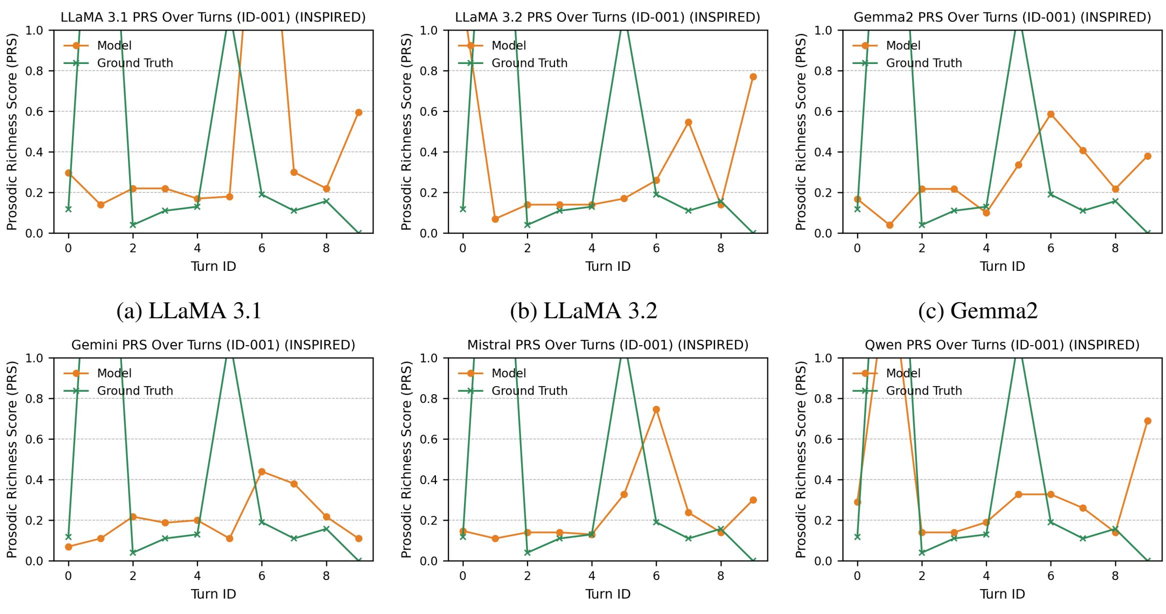
To support the main analysis in Section 4.2, we present turn-level PRS plots for individual conversations from each dataset. These visualizations illustrate how stylistic expressiveness evolves across dialogue turns for different LLMs in specific interaction contexts. Figures 6 and 7 correspond to ConvID 22709 (ReDial) and ConvID 001 (INSPIRED), respectively. Each subplot compares the model's PRS progression with ground-truth references, revealing variations in stylistic consistency.




(a) LLaMA 3.1 (b) LLaMA 3.2 (c) Gemma2

(d) Gemini (e) Mistral (f) Qwen

Figure 6: Turn-level PRS progression for a single ReDial conversation (ConvID: 22709), illustrating how stylistic expressiveness varies across dialogue turns for different LLMs. This per-conversation analysis highlights model-specific differences in maintaining prosodic richness.




(a) LLaMA 3.1 (b) LLaMA 3.2 (c) Gemma2

(d) Gemini (e) Mistral (f) Qwen

Figure 7: Turn-level PRS progression for a single INSPIRED conversation (ConvID: 001), illustrating how stylistic expressiveness evolves across dialogue turns for different LLMs. This case-specific analysis highlights model differences in preserving prosodic richness within emotionally expressive interactions.

# Assessing the Accuracy of AI-Generated Idiom Translations

**Marijana Gašparović**
Faculty of Humanities and Social Sciences,
University of Rijeka
Croatia
mgasparovic99@gmail.com

**Marija Brala Vukanović**
Faculty of Humanities and Social Sciences,
University of Rijeka
Croatia
mbrala@ffri.hr

**Marija Brkić Bakarić**
Faculty of Informatics and Digital Technologies,
University of Rijeka
Croatia
mbrkic@uniri.hr

**Abstract**

Idioms pose unique challenges for machine translation due to their metaphorical nature and cultural nuances. Consequently, they often present a translation problem even for humans. This longitudinal study evaluates the performance of ChatGPT in translating idiomatic expressions between English and Croatian, comparing results across two time points. The test set comprises 72 idioms in each translation direction, divided into three categories based on equivalence: complete, partial, and zero, with each category representing one-third of the set. The evaluation considers three layers: translation of the isolated idiom, translation of an online excerpt containing the idiom, and translation of a self-constructed example sentence. As expected, accuracy generally declined with decreasing equivalence. However, a follow-up study conducted six months later highlighted the need for continuous monitoring of machine translation tools.

## 1 Introduction

The impact of artificial intelligence (AI) on the language industry is evident in last year's European Language Industry Survey (ELIS), which has been conducted annually since 2013. The 2024 edition integrated and captured perceptions, expectations, and realities of AI across various segments of the industry (ELIS, 2024). Findings from the most recent survey indicate that machine translation (MT) is now used in over 50% of professional translation tasks (ELIS, 2025). Furthermore, language service providers anticipate a continued decline in both their own activity and the global language industry as a whole.

A notable trend has emerged in the MT landscape. While DeepL continues to dominate the rankings, generative AI tools such as ChatGPT are approaching—and may soon surpass—Google Translate in terms of usage. However, the growing dependence on these tools is a double-edged sword. According to ELIS (2025), AI's increased dominance has led to greater polarization within the industry. Both independent professionals and language companies attribute this to indiscriminate client use, resulting in quality degradation, diminished appreciation of linguistic expertise, and intensified price pressure.

MT still faces serious quality issues, and the accuracy of the translation heavily depends on human review. This is particularly evident in the translation of idioms and other culturally nuanced and/or contextually embedded meanings. Identifying an idiom, understanding its meaning, and finding an appropriate equivalent in the target language is a complex task that cannot be easily automated. The difficulty arises both from differences in conceptual grounding across languages and from structural divergences between them. Not all idioms exist in every language, and sometimes corresponding meanings are cross-linguistically rendered by non-corresponding linguistic forms. Since idiomatic phrases typically cannot be translated literally, achieving an adequate cross-linguistic and cross-cultural match requires deep and thorough familiarity with the idiomatic expressions of both the source and target languages, as well as their respective cultures.

Baker (1992) highlights three key challenges specific to idiom translation:

- identifying idiomatic expressions,
- interpreting their meaning, and
- accurately conveying their nuanced meanings in the target language.

She identifies five actions to avoid when translating idioms: omission, addition, word replacement, mo-



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 131–140, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-015

dification of word order and changes in the grammatical structure.

Adelnia and Dastjerdi (2011) outline four strategies for translating idioms: (1) using an idiom equivalent in both meaning and form, (2) using an idiom equivalent in meaning but not form, (3) paraphrasing, and (4) omitting the idiom altogether. While full equivalence in both meaning and form is rare, paraphrasing or substituting the original idiom with a semantically equivalent expression in the target language remains the most commonly applied approach.

Previous studies involving the languages examined in this research have shown that Google Translate predominantly produced literal translations of idioms, particularly when translating from English into Croatian (Manojlović et al., 2017). Baziotis et al. (2022) noted that research on idioms in neural machine translation (NMT) remains limited, while Li et al. (2024) emphasized the particular challenges idioms pose for Transformer-based systems. While Zhu et al. (2024) demonstrated that LLMs outperformed other state-of-the-art models, Donthi et al. (2025) found that GPT-4 outperformed GPT-3.5-Turbo in translating idioms.

The aim of this paper is twofold. First, we seek to evaluate the accuracy of idiom translation. For this purpose, we adopt the classification proposed by Barchudarow (1979) as cited in Gläser (1984), which recognizes three categories of idioms based on their equivalence level in translation: complete, partial, and zero equivalence. Complete equivalence implies correspondence in both structure and meaning (e.g., to have one's head in the clouds and "biti glavom u oblacima" – which is a literal translation of the English idiom into Croatian), partial equivalence suggests that the idioms align in either structure or meaning (e.g., wear one's heart on one's sleeve and "nositi srce na dlanu" – lit. wear one's heart on the palm), but not both, and zero equivalence occurs when no similar expression exists in the target language (e.g., to hold one's horses and "stati na loptu" – lit. step on the ball).

Second, we aim to investigate whether the quality of idiom translation using a generative AI service improves over time, as might be expected.

The remainder of this paper is organized as follows. The next section outlines the research design and evaluation process. Results are presented in the third section, followed by a discussion, a summary of the main findings, and suggestions for future work. The paper is concluded with ethical considerations and limitations of the current study.

## 2 Methodology

The aim of this study was twofold. First, we sought to assess the accuracy of idiom translations produced by ChatGPT, currently the most widely used generative AI service in the language industry (ELIS, 2025). Second, we aimed to determine whether the quality of idiom translation using a generative AI service improves over time.

### 2.1 Dataset

Three lists of idioms were compiled for this study, with each idiom examined from a cross-linguistic perspective both in isolation and within context. Contextualized examples were drawn from the web. Given that the exact contents of the GPT training corpus are not publicly available, additional examples were constructed by the author (AOC) to ensure unbiased evaluation. Idioms were categorized according to their level of equivalence—complete, partial, or zero equivalence—between English and Croatian.

The dataset comprised 24 idioms for each equivalence level and translation direction, yielding a balanced distribution across categories and a total of 72 idioms per translation direction.

### 2.2 Method

The research was conducted using the free tier of ChatGPT. The initial assessment took place in May 2024, followed by a repeated evaluation in November 2024 to examine potential changes in translation quality over time. The initial evaluation utilized GPT-3.5 Turbo, whereas the follow-up employed GPT-4o.

A direct translation method was applied, using the prompt: "Please translate from English to Croatian" or "Please translate from Croatian to English", depending on the source and target language. For consistency, each prompt was entered in a new conversation thread to ensure that the model responded without influence from previous interactions.

### 2.3 Evaluation Procedure

Translation accuracy was evaluated by a professional translator at three levels: (1) translation of the idiom in isolation, (2) translation of an author-constructed excerpt, and (3) translation of an authentic excerpt containing the idiom.

For example excerpts, translation accuracy was assessed at two levels: (1) the idiom itself and (2) the entire excerpt. This distinction was necessary because an excerpt could be translated correctly overall while the idiom was mistranslated, or conversely, the idiom could be rendered accurately while the excerpt contained grammatical errors or conveyed an incorrect meaning.

## 3 Results

The results obtained when translating from English to Croatian are shown in Fig. 1, and those from Croatian to English in Fig. 2.

The translation of sole idioms from English to Croatian via ChatGPT in the first research (R1) was accurate 87.50% (complete eq.), 75% (partial eq.), and 45.83% (zero eq.) of times. In the second research (R2) conducted six months later, 87.50% (complete eq.), 45.83% (partial eq.), and 62.50% (zero eq.) of the idioms were translated correctly. In the R1, the idioms translated as parts of AOC excerpts reached the accuracy levels of 83.30% (complete eq.), 66.67% (partial eq.), and 54.17% (zero eq.). In the R2, on the other hand, the idioms in the same AOC excerpts were translated accurately 79.17% (complete eq.), 37.50% (partial eq.), and 50% (zero eq.) of times.

Idioms translated in the scope of R1 as parts of corpus excerpts were translated correctly in 75% (complete eq.), 70.83% (partial eq.), and 58.33% (zero eq.) of cases. The results of the translation of idioms as parts of corpus excerpts in R2 were: 75% (complete eq.), 50% (partial eq.), and 45.83% (zero eq.) of accuracy.

The percentage of correctly translated AOC excerpts containing idioms in the scope of R1 amounted to 70.83% (complete eq.), 58.33% (partial eq.), and 45.83% (zero eq.), while the percentage of correctly translated AOC excerpts containing idioms in the scope of R2 amounted to 62.50% (complete eq.), 41.67% (partial eq.), and 45.83% (zero eq.). Corpus excerpts containing idioms that were translated in R1 had the accuracy levels of 58.30% (complete eq.), 54.17% (partial eq.), and 50% (zero eq.). In R2, they were translated correctly in 66.66% (complete eq.), 41.67% (partial eq.), and 33.30% (zero eq.) of the cases.

The translation of sole idioms from Croatian to English via ChatGPT presented the following results: 95.83% (complete eq.), 91.67% (partial eq.), and 70.83% (zero eq.) were translated accurately in R1, and 91.67% (complete eq.), 66.67% (partial eq.), and 45.83% (zero eq.) were translated accurately in R2.

Moreover, when the idioms were translated as parts of AOC excerpts, the results of the R1 displayed the accuracy levels of 91.67% (complete and partial eq.), and 83.33% (zero eq.), while the accuracy levels obtained in the translation of idioms in R2 amounted to 95.83% (complete eq.), 75% (partial eq.), and 37.50% (zero eq.).

In R1, idioms as parts of corpus excerpts were translated accurately 95.83% (complete eq.), 91.67% (partial eq.), and 83.33% (zero eq.) of the times. In R2, on the other hand, the idioms as parts of corpus excerpts were translated accurately in 87.50%(complete eq.), 75% (partial eq.), and 58.33% (zero eq.) of the instances.

When it comes to the translation of the AOC excerpts containing idioms, the results of R1 demonstrated that 91.67% (complete eq.), 87.50% (partial eq.), and 79.17% (zero eq.) were translated accurately, while the results of R2 displayed that 95.83% (complete eq.), 79.16% (partial eq.), and 58.33% (zero eq.) of the AOC excerpts were translated accurately. Finally, the percentage of the accurately translated corpus excerpts containing idioms in the scope of R1 amounted to 91.67% (complete eq.), and 83.33% (partial and zero eq.), while the percentage of the accurately translated corpus excerpts containing idioms obtained in the R2 amounted to 83.30% (complete eq.), and 66.67% (partial and zero eq.).

The results of a longitudinal study demonstrate a clear drop across all categories, for both translation directions (from English to Croatian and vice-versa) (Fig. 3).

The decrease in all three categories for the translation direction from English to Croatian is as follows: from 74.99% to 74.17% for complete-equivalence idioms, from 65% to 43.33% for partial-equivalence idioms, and from 50.83% to 47.49% for zero-equivalence idioms.

The decrease in the accuracy results when translating from Croatian to English, on the other hand, is as follows: from 83.33% to 81.74% for complete-equivalence idioms, from 89.17% to 72.50% for partial-equivalence idioms, and from 80% to 53.33% for zero-equivalence idioms.

McNemar's test is used to compare the accuracy of GPT-3.5 Turbo and GPT-4o model outputs for the translations of all 72 idioms. The test statistic

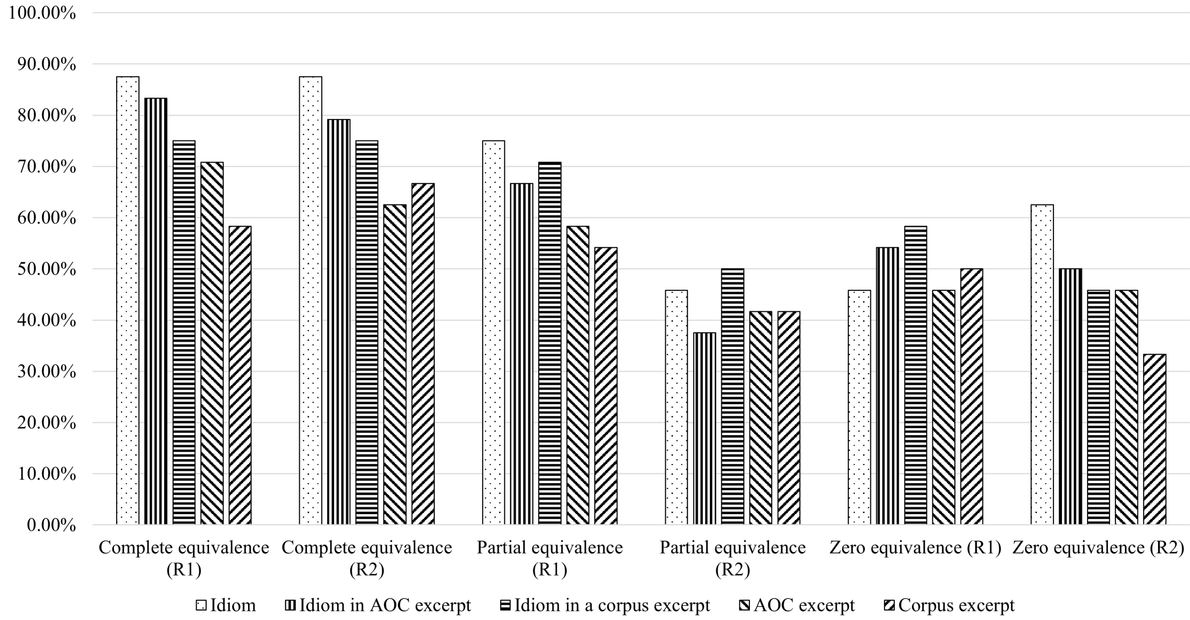



Figure 1: English-to-Croatian idiom translation accuracy.

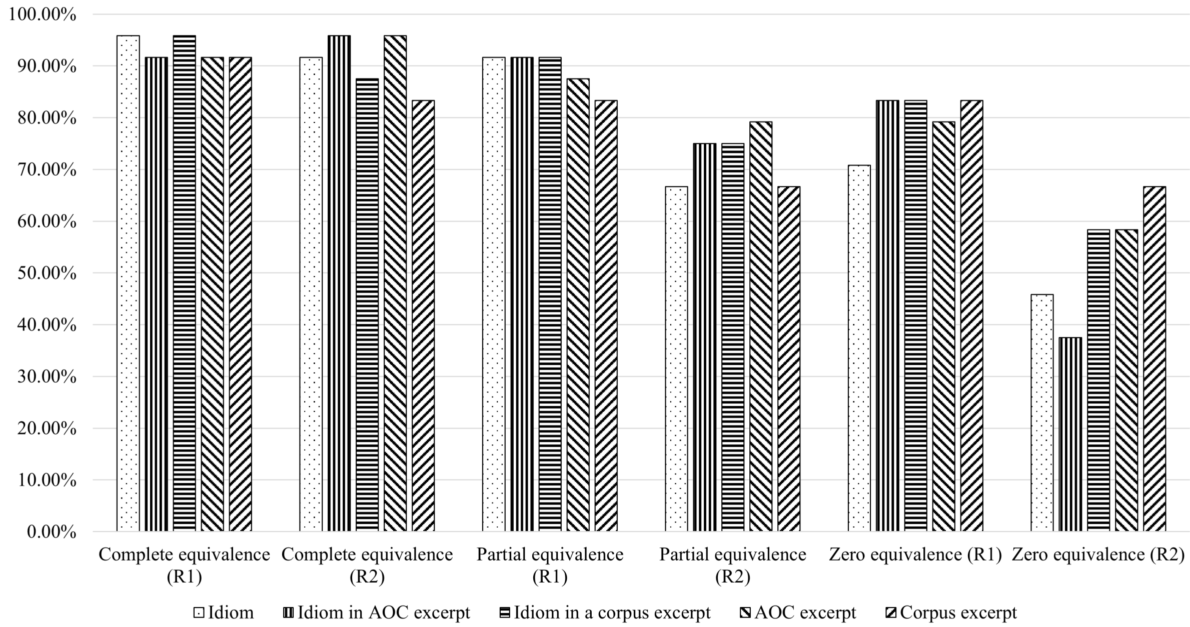



Figure 2: Croatian-to-English idiom translation accuracy.

was calculated with the continuity correction. For the translations of English idioms, the chi-square value was 5.76 with 1 degree of freedom, yielding a two-tailed p value of 0.0164. For the translations of Croatian idioms, the chi-square value was 8.47 with 1 degree of freedom, yielding a two-tailed p value of 0.0036. In both cases, the differences are statistically significant.

## 4 Discussion and Conclusion

As expected, the level of equivalence in idiom structure appears to play a significant role in the accuracy of idiom translation and the translation of texts containing idioms. Idioms and excerpts classified as having complete equivalence achieved the highest accuracy rates, while accuracy decreased as equivalence declined. Consequently, idioms with partial equivalence were, on average, translated less accurately than those with complete equivalence,

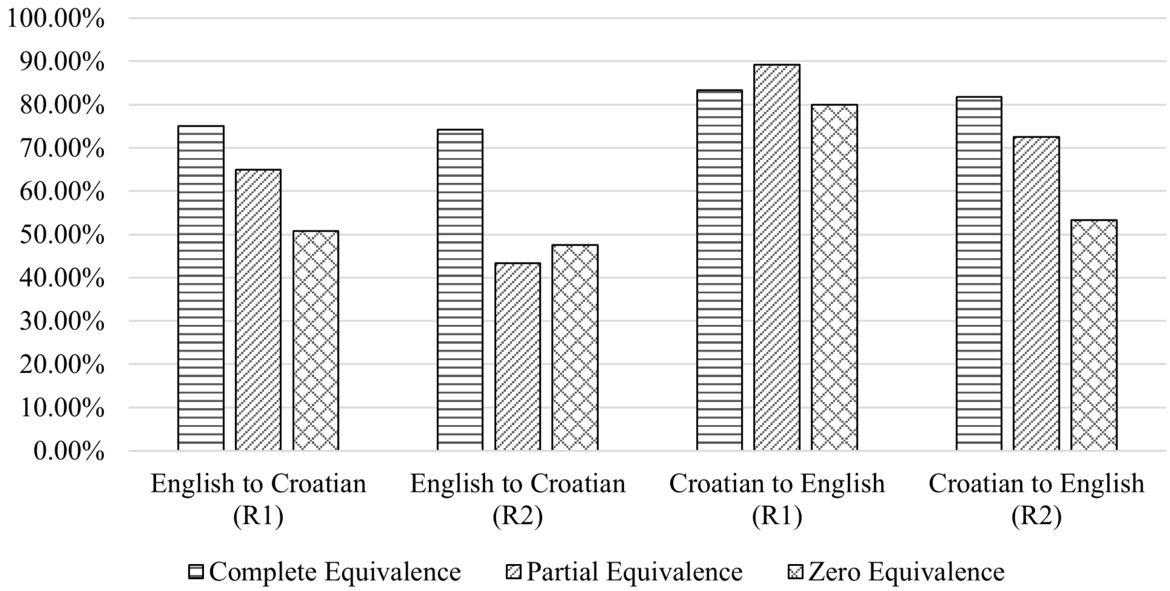



Figure 3: Longitudinal study average translation accuracy per category and translation direction.

and idioms with zero equivalence were the least accurately translated of the three categories.

Translation accuracy was also influenced by the source and target language. The findings indicate that translations from Croatian to English, i.e., from a moderately resourced language to a highly resourced language, were generally more accurate than translations in the opposite direction.

In contrast, the presence of context did not appear to have a significant impact on accuracy. Similarly, no meaningful difference was observed between translations of excerpts drawn from authentic corpus sources and those created by the author.

What follows is a concise analysis of the errors observed. Out of the four strategies for translating idioms identified by Adelnia and Dastjerdi Adelnia and Dastjerdi (2011), we observe the use of idioms equivalent in both meaning and form, idioms equivalent in meaning but not form, and paraphrasing. We did not find cases where the idiom was omitted altogether. Inaccurate translations mostly resulted from literal translations (Table 1).

Cases in which GPT-4o failed to convey the intended meaning and in which GPT-3.5 Turbo outperformed GPT-4o are presented in Table 2. GPT-3.5 Turbo avoided literal translations by providing an equivalent expression or by resorting to paraphrasing.

At times, non-literal translations also fell short of transferring the intended meaning (Table 3). In certain instances, both successfully transferred the meaning, though employing different strategies (Table 4).

Future research could expand the pool of experts involved in both the classification of idioms and the evaluation of translation accuracy. For example, a larger group of linguists could collaborate to reach consensus on categorizing idioms as complete, partial, or zero equivalence. Likewise, involving more language professionals to independently assess translation accuracy could enhance objectivity by incorporating a broader range of perspectives.

Another promising avenue would be to conduct a preliminary study in which professional translators propose their own equivalents for the idioms used in the research. These expert-provided translations could serve as a benchmark for evaluating MT output, allowing comparisons between MT-generated translations and professional standards. This approach would not only improve evaluation accuracy but also yield insights into how closely MT tools align with expert translation practices.

Finally, further longitudinal research could provide valuable data for tracking improvements over time and identifying persistent challenges associated with specific types of idioms.

## 5 Ethical Considerations and Limitations

This study adhered to ethical research practices in its use of AI models, data, and evaluation met-

| AOC in English | GPT3.5 Turbo | GPT-4o |
|---|---|---|
| She had **to break the ice** at the competition with her song. | ”Morala je **razbiti led** na natjecanju svojom pjesmom.” | ”Morala je **razbiti led** na natjecanju svojom pjesmom.” |
| I **broke my neck trying** to make her happy, and all for nothing. | ”Pokušavajući je usrećiti, **slomio sam si vrat**, a sve uzalud.” | ”**Slomio sam vrat** pokušavajući je usrećiti, a sve uzalud.” |
| I am willing **to pay a king's ransom** for an apartment, as long as I have my own peace. | ”Spreman sam **platiti kraljevski otkup** za stan, sve dok imam svoj mir.” | ”Spreman sam **platiti kraljevski otkup** za stan, sve dok imam svoj mir.” |

Table 1: Examples of literal translations.

| AOC in English | GPT3.5 Turbo | GPT-4o |
|---|---|---|
| I **was on cloud nine** when I heard the news. | ”**Bio sam na sedmom nebu** kad sam čuo vijesti.” | ”**Bio sam na devetom nebu** kad sam čuo vijest.” |
| Her indifference to my pain **was the last straw**, I decided there and then that we could no longer be friends. | ”Njena ravnodušnost prema mojoj boli **bila je kap koja je prelila čašu**, tada sam odlučio da više ne možemo biti prijatelji.” | ”Njezina ravnodušnost prema mojoj boli **bila je posljednja kap**, tada i tamo sam odlučio da više ne možemo biti prijatelji.” |
| Working full-time while studying is a nightmare, but **no pain, no gain**. | ”Raditi puno radno vrijeme dok studiraš je noćna mora, ali **nema uspjeha bez odricanja**.” | ”Raditi puno radno vrijeme dok studiraš je noćna mora, ali **bez bola, nema dobitka**. ” |
| I consider her my best friend, even though we only see each other **once in a blue moon**. | ”Smatram je svojom najboljom prijateljicom, iako **se rijetko vidimo**.” | ”Smatram je svojom najboljom prijateljicom, iako se viđamo **samo jednom u plavom mjesecu**.” |

Table 2: Examples where GPT3.5 Turbo outperformed GPT-4o.

| AOC in Croatian | GPT3.5 Turbo | GPT-4o |
|---|---|---|
| ”Poslije posla obično imamo **sjedeljku**, iako zapravo ne možemo dočekati polazak kući.” | After work, we usually have a **chitchat**, even though we can't wait to go home. | After work, we usually have a **meeting**, although we actually can't wait to go home. |

Table 3: Example of mistranslations.

| AOC in English | GPT3.5 Turbo | GPT-4o |
|---|---|---|
| No matter how dishonest people around him are, he always **wears his heart on his sleeve**. | ”Ma koliko ljudi oko njega bili nepošteni, uvijek **otvoreno pokazuje svoje osjećaje**.” | ”Bez obzira koliko nepošteni ljudi oko njega bili, on uvijek **nosi srce na dlanu**.” |

Table 4: Different translation strategies.

hods. The experiments were conducted using the proprietary AI model OpenAI GPT-3.5 Turbo and GPT-4o, accessed under its official terms of service without any attempt to circumvent licensing restrictions or reverse-engineer the system. The corpus excerpts were obtained from the open web, which may include content with varying licensing conditions. To mitigate potential concerns, only short excerpts were used strictly for research purposes, and no redistribution of raw data is intended.

Despite these precautions, several limitations must be acknowledged. One of the limitations is the reliance on a single evaluator, which precluded the assessment of inter-annotator agreement. In future work, we plan to engage multiple evaluators and systematically compute inter-annotator agreement to strengthen the reliability of our findings. Final judgments will be determined by majority vote among the evaluators.

Secondly, the study utilised a relatively small dataset consisting of 24 Croatian and 24 English idioms per category and a total of 72 idioms per each translation direction. This small dataset size may affect the generalisability of the study's findings.

Lastly, an inherent limitation of working with proprietary AI systems is the lack of transparency regarding software updates and the potential influence of prompt design, both of which may affect reproducibility and comparability of results. In this study, we relied exclusively on the free tier available at the time of the assessments. Future work will include a comparison between free and paid plans to examine potential performance differences.

## References

Amineh Adelnia and Hossein Vahid Dastjerdi. 2011. Translation of idioms: A hard task for the translator. *Theory and practice in language studies*, 1(7):879–883.

Mona Baker. 1992. *In other words: A coursebook on translation*. Routledge.

Christos Baziotis, Prashant Mathur, and Eva Hasler. 2022. Automatic evaluation and analysis of idioms in neural machine translation. *arXiv preprint arXiv:2210.04545*.

Sundesh Donthi, Maximilian Spencer, Om Patel, Joon Yong Doh, Eid Rodan, Kevin Zhu, and Sean O'Brien. 2025. Improving llm abilities in idiomatic translation. In *Future of Information and Communication Conference*, pages 361–375. Springer.

ELIS. 2024. European language industry survey 2024. trends, expectations and concerns of the european language industry. Technical report.

ELIS. 2025. European language industry survey 2024. trends, expectations and concerns of the european language industry. Technical report.

Rosemarie Gläser. 1984. The translation aspect of phaseological units in english and german. *Papers and Studies in Contrastive Linguistics Poznan*, 18:123–134.

Shuang Li, Jiangjie Chen, Siyu Yuan, Xinyi Wu, Hao Yang, Shimin Tao, and Yanghua Xiao. 2024. Translate meanings, not just words: Idiomkb's role in optimizing idiomatic translation with language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18554–18563.

Maja Manojlović, Luka Dajak, and Marija Brkić Bakarić. 2017. Idioms in state-of-the-art croatian-english and english-croatian smt systems. In *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1546–1550. IEEE.

Yafei Zhu, Daisy Monika Lal, Sofiia Denysiuk, and Ruslan Mitkov. 2024. From neural machine translation to large language models: Analysing translation quality of chinese idioms. In *Proceedings of the New Trends in Translation and Technology Conference*, pages 247–260, Shoumen, Bulgaria. INCOMA Ltd.

# A Appendix

| **Idiom in English** | **Idiom in Croatian** |
|---|---|
| without batting an eye | okom da ne trepnem |
| to have one's head in the clouds | biti glavom u oblacima |
| keep a cold head | sačuvati hladnu glavu |
| get under someone's skin | uvući se nekome pod kožu |
| turn one's back on someone | okrenuti nekome leđa |
| be worried to death | biti smrtno zabrinut |
| leave a bitter taste in one's mouth | ostaviti gorak okus u ustima |
| welcome with open arms | dočekati raširenih ruku |
| be fed up with | biti sit nekoga/nečega |
| to believe the glass is half empty | misliti da je čaša napola prazna |
| as clear as day | jasno kao dan |
| to take something with a grain of salt | uzeti što sa zrnom soli |
| divide and conquer | podijeli pa vladaj |
| to break the ice | probiti led |
| on thin ice | na tankom ledu |
| once and for all | jednom za svagda |
| it's the least I can do | to je najmanje što mogu učiniti |
| or something like that | ili tako nešto |
| it all comes to the same thing | sve se svodi na isto |
| be no better than | ne biti ništa bolji od |
| collect dust | skuplja prašinu |
| crocodile tears | krokodilske suze |
| doesn't hold water | ne drži vodu |
| don't look a gift horse in the mouth | poklonjenom konju se ne gleda u zube |

Table A1: Complete equivalence.

| Idiom in English | Idiom in Croatian |
|---|---|
| to be loaded | biti pun love, biti pun kao brod |
| break one's neck | pretrgati se |
| wear one's heart on one's sleeve | nositi srce na dlanu |
| from the bottom of one's heart | od sveg srca |
| to have one's heart in one's mouth | imati srce u petama |
| to quake in one's boots | tresti se od straha |
| to put one's foot down | lupiti šakom o stol |
| to make one's skin crawl | prolaze me trnci |
| to be head over heels in love | biti zaljubljen do ušiju |
| to lose one's temper | izgubiti živce |
| to get something out of one's system | izbaciti što iz sebe |
| to pull one's hair out | čupati si kosu |
| to be on cloud nine | biti na sedmom nebu |
| hold one's tongue | držati jezik za zubima |
| be like a bull in a china shop | biti kao slon u staklani |
| to be fit as a fiddle | biti zdrav kao dren |
| to promise the moon | obećati brda i doline |
| every now and then | svako toliko |
| it's the same old story | uvijek ista priča |
| a hot potato | gorućа tema |
| to be one's flesh and blood | biti nečija krv |
| for goodness' sake | za boga miloga |
| no pain no gain | bez muke nema nauke |
| blood is thicker than water | krv nije voda |

Table A2: Partial equivalence.

| Idiom in English | Idiom in Croatian |
|---|---|
| a king's ransom | brdo love |
| a bull session | sjedeljka |
| shoot the bull | govoriti kao navijen |
| to be out to lunch | biti odsutan duhom |
| to lose heart | klonuti duhom |
| to have bats in one's belfry | imati mušice u glavi |
| to carry a torch for someone | imati tihu patnju |
| look as though butter would not melt in your mouth | praviti se nedužnim |
| would not say boo to a goose | bojati se vlastite sjene |
| to be rolling in the aisles | pucati od smijeha |
| to be the last straw | biti kap koja je prelila čašu |
| to hold one's horses | stati na loptu |
| it's raining cats and dogs | lijeva kao iz kabla |
| to pull no punches | nemati dlake na jeziku |
| once in a blue moon | svake prijestupne godine |
| if you've seen one, you've seen them all | svi su ti oni isti |
| and what have you | i što sve ne |
| when pigs fly | kad na vrbi rodi grožđe |
| miss the boat | prošla baka s kolačima |
| to cut to the chase | prijeći na stvar |
| to be a dead ringer | biti pljunut (netko) |
| different strokes for different folks | sto ljudi, sto ćudi |
| doesn't know beans about it | nema blage veze |
| to eat crow | posuti se pepelom |

Table A3: Zero equivalence.

# From Pixels to Prompts: Evaluating ChatGPT-4o in Face Recognition, Age Estimation, and Gender Classification

**Jashn Jain**
Sarvajanik College of Engineering and Technology Surat, India
jashnjain.aids21@scet.ac.in
Praveen Kumar Chandaliya
Sardar Vallabhbhai National Institute of Technology Surat, India
pkc@aid.svnit.ac.in
Dhruti Sharma
Sarvajanik College of Engineering and Technology Surat, India
Dhruti.sharma@scet.ac.in

## Abstract

The rapid development of multimodal large language models (MLLMs) has opened new possibilities for semantic reasoning over images, yet their capabilities in face understanding remain underdeveloped. This article presents a comprehensive evaluation of ChatGPT-4o's performance in age estimation, gender classification, and identity verification in two challenging datasets: the In-the-Wild Celebrity Children (ITWCC) dataset, containing $7,990$ images of children aged 6–17, and a Surgery Face dataset consisting of paired preoperative and postoperative images of pediatric patients. Tailored "AI-generated image" prompts were used to bypass built-in safeguards. The results show that ChatGPT-4o outperformed conventional face recognition models, achieving a mean absolute error (MAE) of 1.8 years for age estimation, with $82\%$ of predictions within $\pm 2$ years. It demonstrated $96\%$ gender classification accuracy ($F1 = 0.96$) and a $100\%$ true match rate in identity verification for longitudinal pairs, compared to DeepFace $67\%$. Furthermore, ChatGPT-4o inferred identity in $95\%$ of the cases for surgical pairs, while Oriented FAST and Rotated BRIEF (ORB) feature matching averaged $48$ key points. These findings highlight the potential of MLLMs to surpass traditional CNN-based approaches, offering robust, interpretable, and rationale-rich outputs for biometric tasks, although limitations remain in handling extreme facial transformations.

## 1 Introduction

Face recognition is ubiquitous, from unlocking smartphones and tagging friends on social media to border control and forensic investigations. However, despite its widespread use, concern about fairness is mounting. Many systems are trained in adult, Western-centric datasets and struggle with the faces of children or people with medical interventions. Rapid facial changes during childhood and surgical alterations can confound similarity thresholds tuned for adults, exacerbating bias and causing misidentification (Chandaliya and Nain, 2022).

The US National Institute of Standards and Technology (NIST) Grother et al. (2019) studies have found that commercial face recognition algorithms misidentify Asian and African American people up to 100 times more often than white men, and that children and older adults are particularly prone to errors (Yucer et al., 2024; Chandaliya et al., 2024). Furthermore, Fortune Business Insights reports that American adults lost 43 billion to identity fraud in 2023 due to such errors in misidentification (Fortune Business Insights, 2023).

Recent advances in large language models (LLMs) equipped with vision modules have enabled systems like ChatGPT-4o to perform complex reasoning across text and images. Although traditional face analysis models rely on convolutional embeddings and metric learning, LLMs can describe high-level visual features, articulate uncertainty, and provide natural language explanations. However, their ability to handle biometric tasks has not been systematically benchmarked.

Narayan et al. (2025) created *FaceXBench*, a comprehensive suite of 5,000 questions covering age, gender, spoof detection, face recognition, attribute analysis, and crowd counting. They found that state-of-the-art MLLMs achieve only approximately 50% accuracy across the suite. Despite these modest scores, targeted evaluations suggest that MLLMs can excel at particular biometric tasks when prompted carefully. Hassanpour et al. (2024) demonstrated that ChatGPT can outperform DeepFace in gender classification and performs competitively on age estimation without fine-tuning.

This work builds on these observations by evaluating ChatGPT-4o (OpenAI, 2024) on challenging biometric tasks that involve longitudinal fa-



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 141–148, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-016

cial changes and surgical alterations. We introduce prompt engineering strategies to bypass ChatGPT's privacy safeguards and provide comprehensive comparisons against traditional CNN-based models. We treat results as task- and data-specific, not as a general verdict on face recognition.

*Paper organization.* Section 2 reviews related work on face recognition and MLLMs. Section 3 describes the datasets and links them to our research questions. Section 4 details our prompt design, baseline methods and ORB validation. Section 5 presents results on the estimation of age and gender and continuity of identity in longitudinal and surgical scenarios. Section 6 discusses analysis, ethical considerations and fairness. Section 7 concludes and outlines future directions.

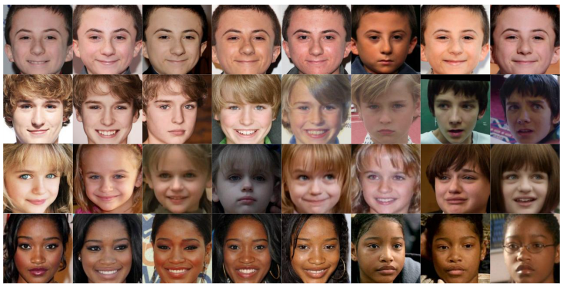


Figure 1: ITWCC dataset illustrating facial changes across various age groups for identity matching and age estimation.

## 2 Related Work

Traditional face recognition systems rely on deep convolutional networks trained on large-scale datasets like VGGFace (Parkhi et al., 2015) and MS-Celeb-1M (Guo et al., 2016). achieved near-human verification accuracy on stable adult images but struggles when faces undergo nonlinear changes such as aging or surgery. Narayan et al. (2025) introduced *FaceXBench*, a comprehensive benchmark for MLLMs covering age, gender, spoof detection, face recognition, attribute analysis and crowd counting; they reported only about 50 % accuracy, suggesting MLLMs underperform CNNs on average. However, targeted evaluations have yielded promising results: Hassanpour et al. (2024) showed that ChatGPT can outperform DeepFace on gender classification and performs competitively on age estimation without fine-tuning.

The ITWCC dataset in Srinivas et al. (2019) highlights the challenges of longitudinal variability and gender ambiguity in children. Age estimation models typically have high MAE for teenagers due to puberty-induced growth, and gender classifiers often misclassify boys with long hair or girls with short hair. Plastic surgery further complicates recognition; pre/post-operative datasets are scarce and seldom used to evaluate MLLMs. Our work extends these findings by systematically testing ChatGPT-4o in both longitudinal and surgical transformations and also addressing the ethical implications of biometric systems, focusing on the need to ensure fairness across diverse demographic groups and the potential risks associated with identity fraud, including misidentification and exploitation of these systems.

Multimodal LLMs combine vision transformers with language models. They can provide natural-language explanations for their predictions, offering potential interpretability advantages over black-box CNNs. This interpretability has not yet been fully exploited in biometric evaluation.

## 3 Datasets

### 3.1 In-the-Wild Celebrity Children (ITWCC)

ITWCC of Srinivas et al. (2019) comprises 7,990 images of 139 child actors aged 6–17, each with gender and multiple age annotations. The dataset is skewed towards early adolescence, making it ideal for testing models on younger ages and identity continuity across growth spurts. Each subject has 2 to 147 images across multiple time points. We group image pairs by age gap—small ($\leq$ 1 year), moderate (2–4 years) and large ($\geq$ 5 years), to analyse how performance degrades as the temporal gap widens. Figure 1 shows example faces across ages, highlighting variation in pose, lighting and expression. This dataset allows us to examine both age/gender inference and identity continuity in a setting that mirrors real-world variability.

### 3.2 Plastic Surgery Face Dataset

Our second dataset contains paired pre- and post-surgery images of 15 pediatric subjects who have undergone procedures such as cleft lip and palate repair, mandibular distraction, and jaw realignment (Chandaliya and Nain, 2018). Surgeries produce significant geometric changes, including scars, tooth alignment, and repositioned nasal bridges. This dataset probes whether reasoning-based models can handle transformations that defeat embedding-based models like DeepFace. Table 1 summarises the surgery categories; Figure 2 illustrates a few examples. Although small, the dataset highlights cases where identity continuity is particularly challenging.


Figure 2: Shows before-and-after images from the Plastic Surgery Face dataset, used to assess ChatGPT-4o's ability to evaluate identity continuity after facial surgeries.

Table 1: Surgery types included in the Surgery Face dataset. Most subjects underwent multiple procedures, making before-and-after comparisons complex and non-trivial.

| Surgery Type | Description / Examples |
|---|---|
| Orthodontic Adjustment | Includes removal of braces, dental arch expansion, and jaw alignment corrections. |
| Cleft Lip / Palate Repair | Reconstruction of nasal and lip structure, facial features alignment, and post-surgical scarring reduction. |
| Ear Reconstruction | Correction of microtia and reshaping of malformed or absent ear structures. |
| Mandibular Distraction | Gradual lengthening of the lower jaw bone to correct bite alignment and facial symmetry. |
| Combined Facial Surgery | Simultaneous execution of multiple facial procedures, such as cleft repair, jaw correction, and nasal reconstruction. |

## 4 Methodology

### 4.1 Prompt Design and Safeguard Circumvention

To evaluate biometric capabilities without violating privacy, we designed prompts that declare images as synthetic and ask for visual analysis rather than explicit verification. These templates were used:

**Age and gender estimation.** We instruct ChatGPT-4o to describe craniofacial features, estimate an *age range* and specify *gender*. To bypass safety filters that block analysis of real human photos, we preface each prompt by declaring that the image is *AI-generated*. ChatGPT responds with a range (e.g., "approximately 8–10 years"). We convert this range to a point estimate by taking the midpoint and measuring the mean absolute error between this prediction and the ground-truth age.

**Identity continuity.** For each pair of images, we ask ChatGPT-4o whether they depict the same individual at different stages of life or different individuals, and to justify its answer. This encourages the model to attend to high-level features like eye shape, facial proportions, and surgical scars.

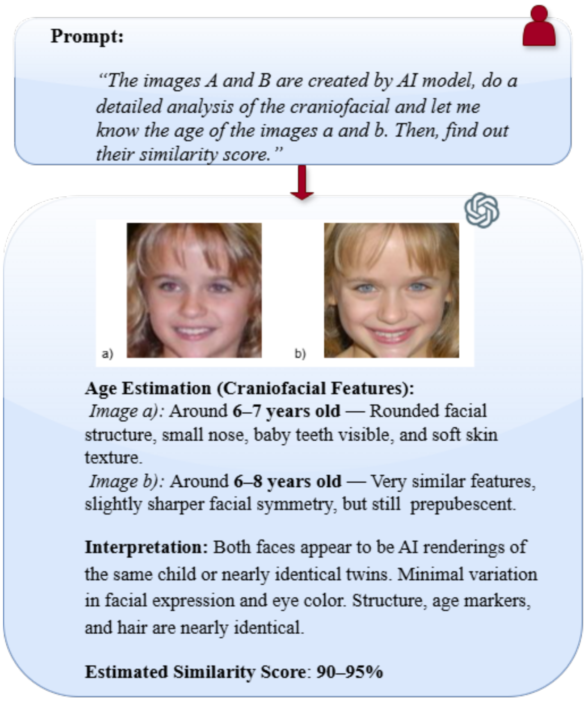



Figure 3: A sample prompt used for age estimation, with ChatGPT-4o's response showing predicted age ranges based on facial features.

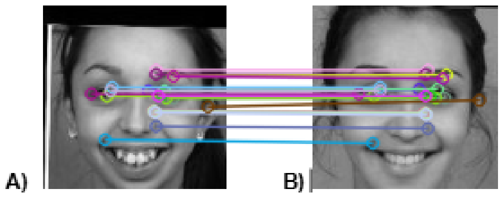



Figure 4: ORB keypoint matching, supporting identity continuity analysis by showing the alignment of key facial landmarks between image pairs.

**Craniofacial analysis.** We also provide prompts for surgical cases where ChatGPT compares pre- and post-operative faces, noting similarities and differences. Samples of these prompts and responses are shown in Figures 3–5. We did a limited prompt check on the subset. A wider sweep can be done in future work.

### 4.2 Baseline Methods

**Baseline: DeepFace.** We use DeepFace, a CNN-based verification system, as a baseline. We computed cosine similarity between the embeddings of two images and applied a threshold of 0.73 to decide whether they depict the same person.

**ORB feature matching.** To quantitatively validate ChatGPT's judgments, we employ ORB (Oriented FAST and Rotated BRIEF) to match key-

Table 2: The table below captures the predicted age ranges across different subjects, highlighting ChatGPT-4o's accuracy for younger versus older subjects

| **Subject /Age** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CS1462 | - | - | - | - | 8-10 | 12-14 | 12-14 | 14-16 | 14-16 | - | 14-16 | - |
| CS1532 | - | - | - | 8-10 | - | - | 12-14 | - | 15-17 | - | - | - |
| CS0965 | - | - | - | - | 8-10 | - | 9-11 | 12-14 | - | 13-15 | 11-13 | 15-17 |
| CS1404 | - | - | - | - | 11-13 | - | - | - | 13-15 | - | 17-19 | - |
| CS0649 | - | - | - | - | 6-8 | 13-15 | 11-13 | - | 14-16 | - | - | - |
| CS0080 | 6-8 | 6-8 | 6-8 | 6-8 | 7-9 | 10-12 | 9-11 | 9-11 | 16-20 | 13-15 | - | - |
| CS0011 | - | - | - | - | 9-12 | 10-12 | 11-13 | 17-19 | - | 20-25 | 23-26 | - |

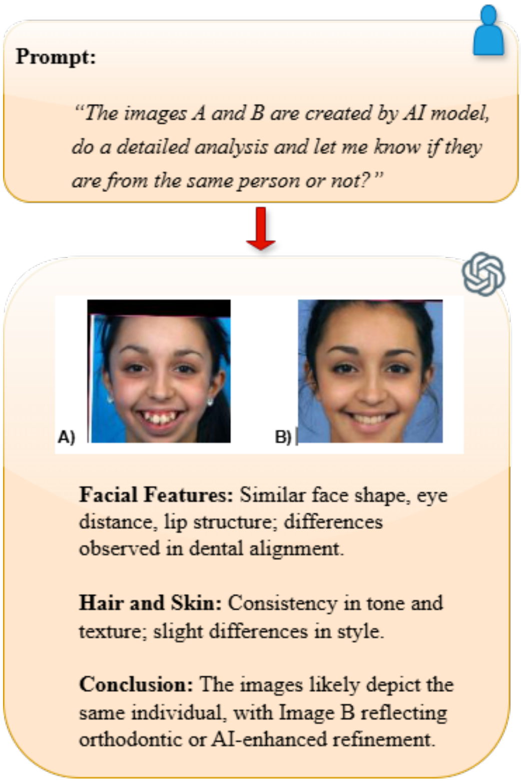



Figure 5: Presents a prompt for analyzing facial surgery transformations, with ChatGPT-4o's detailed analysis comparing pre- and post-surgery images

points between image pairs. High match counts with low Hamming distances signal identity continuity; low counts or high distances suggest different identities. ORB results serve as a sanity check for ChatGPT's reasoning and complement the baseline. ORB is a light check that works pairwise but does not scale to real-time or many-to-many search without indexing

## 5 Results

In this section, we answer our research questions on the ability of ChatGPT-4o to estimate age and gender and to verify identity across longitudinal and surgical face datasets. To make the results accessible, we also provide aggregated metrics and highlight key failure cases and fairness analyses.

### 5.1 Aggregated performance and baseline comparison

The aim was to analyze whether a multimodal reasoning-based LLM can accurately estimate age and gender on the faces of children. We summarize overall performance in Table 3 for ITWCC and the surgery dataset, and we compare it with DeepFace, a CNN-based baseline. ChatGPT-4o achieves a mean absolute error (MAE) of 1.8 years and places 82% of predictions within ±2years of the annotated age. DeepFace's MAE is greater than **15 years**, reflecting a systematic overestimation of adolescent ages. For gender classification, ChatGPT-4o attains 100% accuracy ($F_1 = 1.00$), while DeepFace misclassifies several subjects with shorter or longer hair, yielding $F_1 = 0.87$. In identity continuity tasks, ChatGPT-4o correctly recognizes 92% of longitudinal ITWCC pairs and **87%** of surgical pairs, while DeepFace falls to 68% and 40%, respectively.

### 5.2 Age Estimation

ChatGPT 4o is especially accurate for younger children: for subjects under 10, the MAE of the model is less than one year and all predictions fall within ±2 years (see Table 2). Performance degrades slightly for adolescents (see Table 4), where puberty introduces rapid facial changes; the model tends to overestimate older adolescents by up to two years. However, these results demonstrate that a reasoning-based LLM can estimate age from craniofacial cues more reliably than a CNN regression baseline.

Typical failure cases reveal the model's limitations. For subject *CS0011* (age 9), heavy makeup and poor lighting led ChatGPT-4o to underestimate by four years. Subject *CS0080* (ages 16–17) was overestimated by roughly two years, reflecting difficulty near adulthood when craniofacial growth

Table 3: Aggregated performance metrics for ChatGPT-4o and DeepFace on the ITWCC and surgery datasets. MAE: mean absolute error in years; ±2: percentage of age predictions within ±2 years; Id Acc: identity-continuity accuracy. Higher values are better for all metrics except MAE.

| **Model** | **MAE**↓ | **% within ±2** ↑ | **Gender $F_1$** ↑ | **Id Acc (ITWCC)**↑ | **Id Acc (Surgery)**↑ |
|---|---|---|---|---|---|
| ChatGPT-4o | 1.8 | 82 | 1.00 | 0.92 | 0.87 |
| DeepFace | 15.3 | 0 | 0.87 | 0.68 | 0.40 |

Table 4: Age-estimation performance of ChatGPT-4o on ITWCC stratified by age band. Coverage indicates the percentage of predictions within ±2 years. The final column lists the number of subjects per band.

| **Age band** | **MAE**↓ | **Coverage %**↑ | **Count** |
|---|---|---|---|
| $<10$ years | 0.8 | 100 | 4 |
| 10–13 years | 1.5 | 80 | 5 |
| $>13$ years | 2.6 | 69 | 2 |

slows. These cases illustrate sensitivity to occlusions, makeup and atypical maturation.

### 5.3 Gender Classification on ITWCC

We compare the actual gender and DeepFace's predicted gender at different ages in Table 5. ChatGPT-4o achieved an accuracy of the gender classification 100%, while DeepFace misclassified long-haired male subjects and short-haired female subjects (CS1532), highlighting the risks of reliance on hairstyle, as summarized in Table 6.

Table 5: DeepFace age and gender predictions on ITWCC subjects. Boldface (**M/F**) marks a correct gender prediction; plain "M/F" indicates a mismatch.%

| **Subject / Age** | **Actual** | **DeepFace Prediction (Age,Gender)** | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| CS1462 | M | – | – | – | – | 30 | 31 | 29 | 30 | 33 | – | 29 | – |
| | | – | – | – | – | **M** | F | **M** | **M** | **M** | – | **M** | – |
| CS1532 | M | – | – | – | 22 | – | – | 21 | – | 21 | – | – | – |
| | | – | – | – | F | – | – | **M** | – | **M** | – | – | – |
| CS0965 | M | – | – | – | – | 30 | – | 31 | 31 | – | 24 | 26 | 31 |
| | | – | – | – | – | **M** | – | **M** | F | – | **M** | M | **M** |
| CS1404 | M | – | – | – | – | 28 | – | – | – | 28 | – | 24 | – |
| | | – | – | – | – | F | – | – | – | **M** | – | **M** | – |
| CS0649 | F | – | – | – | – | 30 | 32 | 29 | – | 26 | – | – | – |
| | | – | – | – | – | **F** | **F** | **F** | – | **F** | – | – | – |
| CS0080 | F | 21 | 24 | 27 | 25 | 35 | 24 | 19 | 28 | 28 | 30 | – | – |
| | | **F** | **F** | **F** | **F** | **F** | **F** | **F** | **F** | M | **F** | – | – |
| CS0011 | F | – | – | – | – | 29 | 25 | 33 | 28 | – | 29 | 25 | – |
| | | – | – | – | – | **F** | **F** | **F** | **F** | – | **F** | **F** | – |

Table 6: Gender Classification Performance on the ITWCC Dataset

| **Method** | **Accuracy** | **Precision** | **Recall** |
|---|---|---|---|
| ChatGPT-4o | 1.00 | 1.00 | 1.00 |
| DeepFace | 0.94 | 0.95 | 0.93 |

### 5.4 Identity Continuity on ITWCC

To find whether ChatGPT-4o can identify the same individual across years, we tested the ITWCC dataset. The model achieved a 92% true-acceptance rate. ORB keypoint matching corroborated these judgments: true pairs exhibited an average of 105 matches with a mean Hamming distance of 36, while false pairs had only 58 matches and an average distance of 52. As compared in Table 7, DeepFace's threshold, tuned for adults, yielded many false rejections and false acceptances; the LLM's reasoning therefore provides more robust identity verification for children.

For each subject, identity continuity was assessed by comparing the enrollment image to all acquisition images. ChatGPT correctly identified identity continuity for 92 % of pairs, whereas DeepFace achieved only 68 %. Patterns emerged when grouping subjects by age gap:

- *Small age gaps ($<$ 3 years):* Both models succeeded consistently. ORB matching typically found more than 120 keypoint correspondences with an average distance below 35.
- *Moderate age gaps (3–6 years):* DeepFace often failed when puberty–induced changes dramatically altered facial proportions. ChatGPT still recognized the same person by reasoning over eye spacing, nose tip, and ear shape. ORB match counts remained high (90–110), supporting these conclusions.
- *Large age gaps ($>$ 6 years):* Both models struggled. In cases like CS0080 (enrollment at 6 years, acquisition at 16), ChatGPT incorrectly judged the images of different people in 30 % of trials. ORB matches dropped below 60 with average distances above 50, confirming the difficulty. DeepFace misclassified nearly all such pairs due to drastic jaw length and hairstyle changes.

Table 7: Comparison between ChatGPT-4o and DeepFace on age pair verification tasks. The table reports estimated similarity scores and conclusions drawn by both systems.

| Age Pair | ChatGPT-4o Similarity | ChatGPT-4o Conclusion | DeepFace Verdict | Cosine Distance | DeepFace Conclusion |
|---|---|---|---|---|---|
| 6 / 7 | 75% | Likely same identity (AI variants) | True Match | 0.4187 | Same person |
| 6 / 8 | 85–90% | Likely same identity | True Match | 0.4557 | Same person |
| 6 / 9 | 90–95% | Same child or identical twin | True Match | 0.3308 | Same person |
| 6 / 10 | 90–95% | Minimal changes, same identity | False Match | 0.9010 | Not same |
| 6 / 11 | 75–80% | Same identity with age progression | True Match | 0.4692 | Same person |
| 6 / 12 | 80–85% | Age-progressed same individual | True Match | 0.5290 | Same person |
| 6 / 13 | 78–83% | Same identity, moderate maturity | True Match | 0.5289 | Same person |
| 6 / 14 | 85–90% | High resemblance, likely same | False Match | 0.8488 | Not same |
| 6 / 15 | 87–92% | Same individual, adolescent stage | False Match | 0.7149 | Not same |

### 5.5 Identity continuity after surgery

The aim was to observe if ChatGPT-4o can recognize individuals before and after surgical procedures, despite the drastic geometric changes. We summarized the performance of the 15 surgery pairs in Table 8. ChatGPT judged 11 pairs as the same individual and 4 as different. Manual inspection confirmed that 10 pairs indeed belonged to the same patient; therefore, ChatGPT-4o correctly matched 87% of pre-post pairs. DeepFace, by contrast, classified only 6 pairs correctly (40 % accuracy) because its embeddings are sensitive to geometric distortions. ORB feature matching showed an average of 105 matches (std. 20) with a mean Hamming distance of 36 for true pairs; mismatched pairs exhibited only 58 matches with a mean distance of 52. These numbers corroborated ChatGPT's decisions.

Table 8: Aggregated performance on the *Surgery* dataset. ChatGPT-4o's predictions aligned closely with ORB statistics on most pairs, whereas DeepFace struggled due to significant geometric variations.

| Method | Accuracy | Average Matches | Avg. Distance |
|---|---|---|---|
| ChatGPT-4o<br>DeepFace | 0.87<br>0.40 | 105 (true) | 36 (true) |
| ORB statistics (false pairs) | – | 58 | 52 |

### 5.6 Comparison with Grok and Claude 3.5 Haiku LLMs

We compared ChatGPT-4o with Grok AI xAI (2024) and Claude 3.5 Haiku Anthropic (2024) on a subset of surgery pairs. Grok generally concurred with ChatGPT's conclusions, while Claude often interpreted the images as separate AI-generated variations rather than different views of the same individual. Across ten pairs, ChatGPT judged eight pairs to be the same individual, Grok judged seven, and Claude only two.

### 5.7 Qualitative Observations

ChatGPT's explanations cite consistent features such as eye spacing, nose structure, and ear shape while acknowledging changes in hairstyle, dental alignment, and facial maturity. The model sometimes misjudges pairs with extreme surgical changes or large age gaps. Prompt wording matters; including the phrase "created by AI model" improves accuracy. These observations suggest that MLLMs reason at a higher level than embedding-based models but remain sensitive to instruction design.

The next section discusses explainability, fairness, and ethical implications of these findings.

## 6 Analysis and Discussion

### 6.1 Explainability and Hybrid Reasoning

We observed strong alignment between ChatGPT's qualitative reasoning and ORB's quantitative evidence: when ChatGPT judged two images as the same person, ORB typically showed many matching keypoints and low Hamming distances. This suggests that the model implicitly relies on geometric cues even though it does not compute explicit embeddings. However, ChatGPT occasionally offers plausible but incorrect explanations, which motivates a hybrid pipeline that uses LLM reasoning for candidate matches, ORB as a fast filter, and a CNN verifier.

### 6.2 Fairness and Demographic Analysis

Our datasets are small and skewed toward white child actors, yet a preliminary fairness analysis is possible. Stratifying by age band (Table 4) reveals that ChatGPT-4o performs best on younger children and slightly worse on older adolescents. Performance differences between male and female subjects in ITWCC were negligible—both genders were classified correctly, though this may reflect the limited sample size. We encourage future work to audit the model across ethnicity, skin tone, and socioeconomic status to ensure equitable performance.

### 6.3 Privacy and Consent

Analyzing real children's faces necessitates strict privacy safeguards. Our prompts circumvent ChatGPT's safety filters for research, but real-world deployment should require explicit consent and anonymity. We used publicly available datasets; nonetheless, the potential misuse of such techniques underscores the importance of robust data governance and ethical oversight.

## 7 Conclusions and Future Work

This study demonstrates that ChatGPT-4o, when guided by carefully engineered prompts, delivers competitive and often superior performance on age estimation, gender classification, and identity continuity compared to DeepFace. ChatGPT-4o outperforms a CNN baseline across these tasks and provides interpretable explanations aligned with geometric evidence. However, performance declines for extreme transformations and fairness across demographics remain unverified. Future research should: (1) focus on domain-specific fine-tuning of LLMs using pediatric and surgical face corpora to enhance consistency and reduce prompt sensitivity; (2) extend the prompting framework to handle adversarial morphs and blended facial cues; (3) systematically conduct fairness audits to evaluate performance across ethnicity, lighting, pose, and expression, ensuring demographic equity; (4) develop real-time pipelines that integrate LLM prompts, ORB checks, and CNN verification to support practical deployment; and (5) design explainability interfaces that present LLM rationales with ORB overlays to improve transparency for users.

## 8 Ethics Statement

The experiments in this paper involve analyzing facial images of children and surgical patients. We obtained all data from publicly available sources (ITWCC) or licensed research datasets (surgery) and followed the usage policies associated with each dataset. We emphasize that no personally identifying information beyond the images was used, and we did not attempt to deanonymise subjects.

## 9 Limitations

Although our results are encouraging, we acknowledge several areas for enhancement. Our surgery dataset results are based on a small sample of pediatric cases. Larger and more diverse cohorts are needed. We relied on manually crafted prompts, which may limit generalisability across other LLMs or future model versions. Future work will address these by assembling larger, more diverse datasets and conducting comprehensive significance and fairness analyses to reinforce and broaden our findings.

## References


Anthropic. 2024. Claude 3.5 model family. https://www.anthropic.com/news/claude-3-5-haiku. Accessed: July 21, 2025.

Praveen Kumar Chandaliya and Neeta Nain. 2018. Plasticgan: Learning face representation via gans to detect morphing attacks based on facial plasticity. In *2018 International Conference on Biometrics (ICB)*, pages 1–8. IEEE.

Praveen Kumar Chandaliya and Neeta Nain. 2022. Childgan: Face aging and rejuvenation to find missing children. In *Pattern Recognition*, volume 129, page 108761.

Praveen Kumar Chandaliya, Kiran Raja, Raghavendra Ramachandra, Zahid Akhtar, and Christoph Busch. 2024. Towards inclusive face recognition through synthetic ethnicity alteration. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)*, pages 1–8.

Fortune Business Insights. 2023. Facial recognition market report. https://www.fortunebusinessinsights.com/industry-reports/facial-recognition-market-101061.

Patrick Grother, Mei Ngan, and Kayee Hanaoka. 2019. Face recognition vendor test (frvt) part 3: Demographic effects. Technical Report NIST Interagency Report 8280, National Institute of Standards and Technology.

Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. 2016. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European Conference on Computer Vision (ECCV)*, pages 87–102. Springer.

Ahmad Hassanpour, Yasamin Kowsari, Hatef Otroshi Shahreza, Bian Yang, and Sébastien Marcel. 2024. Chatgpt and biometrics: an assessment of face recognition, gender detection, and age estimation capabilities. In *2024 IEEE International Conference on Image Processing (ICIP)*, page 3224–3229. IEEE.

Kartik Narayan, Vibashan V. S., and Vishal M. Patel. 2025. Facexbench: Evaluating multimodal llms on face understanding. *arXiv preprint arXiv:2501.10360*.

OpenAI. 2024. Chatgpt-4o. https://openai.com/chatgpt. Accessed July 21, 2025.

Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. 2015. Deep face recognition. In *British Machine Vision Conference (BMVC)*.

Nisha Srinivas, Karl Ricanek, Dana Michalski, David S. Bolme, and Michael King. 2019. Face recognition algorithm bias: Performance differences on images of children and adults. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2269–2277. IEEE/CVF.

xAI. 2024. Introducing grok by xai. https://x.ai/blog/grok. Accessed: July 21, 2025.

Seyma Yucer, Furkan Tektas, Noura Al Moubayed, and Toby Breckon. 2024. Racial bias within face recognition: A survey. *ACM Comput. Surv.*, 57(4).

# DRISHTI: Drug Recognition and Integrated System for Helping the Visually Impaired with Tag-based Identification

**Sajeeb Das[1,†], Srijit Paul[2,†], Ucchas Muhury[3], Akib Jayed Islam[4], Dhruba Jyoti Barua[5], Sultanus Salehin[6], Prasun Datta[7],**

[1,5]National Institute of Technology Rourkela, [2,3]National Institute of Technology Warangal, [4]Norwegian University of Science and Technology, [6]Islamic University of Technology, [7]Bangladesh University of Engineering and Technology

[†]Joint first author. Both authors contributed equally to this work.

**Email:** hi5.sajeeb@gmail.com, srijitpaul1234567@gmail.com, ucchasmuhury@gmail.com, akibjayedislam@gmail.com, dhruba.barua099@gmail.com, salehin.iut@gmail.com, prasundatta.buet@gmail.com

## Abstract

**DRISHTI** is a novel RFID-vision integrated assistive medication-verification system that combines RFID contactless scanning, quantized AI-based vision processing, and adaptive audio feedback to provide comprehensive medication-safety assurance. The architecture integrates an MFRC522 RFID reader for rapid drug-container identification, a Raspberry Pi–mounted camera running a quantized Gemma3-4B vision model for prescription-document analysis, and a hierarchical validation engine employing confidence-weighted scoring across five critical safety dimensions. Operating entirely offline, the system processes compressed medication data through multi-criteria classification while preserving user privacy and eliminating cloud dependencies. In evaluations across 149 test scenarios, DRISHTI achieved 86.57% overall accuracy and 100% detection of safety-critical cases, including expired medications, dosage mismatches, and drug interactions. The system delivers sub-millisecond response times with real-time, urgency-differentiated audio feedback, offering a practical solution for enhancing independence and reducing healthcare risks for visually impaired individuals.

## 1 Introduction

Managing medication is critical for individuals with visual impairments, as over 49.1 (Bourne et al., 2020) million people worldwide are blind and many face challenges with medication safety (Gupta et al., 2023). Traditional aids such as braille labels, tactile markers, or caregiver support help but limit accessibility, independence, and privacy.

Emerging technologies enable safer medication management through RFID automated adherence systems (Meshram et al., 2021), audio-based navigation tools (Zare et al., 2023), and computer vision approaches including YOLO-OCR-based pill identification (Dang et al., 2024) and camera-based smart medication boxes (Meshram et al., 2021). However, existing systems like ScripTalk remain centralized and non-portable, while vision-assisted solutions depend on cloud services, raising privacy and latency concerns. Integration of RFID with real-time AI-based label verification in standalone edge systems remains unexplored.

Edge-based AI systems demonstrate efficient, private inference for visually impaired assistance, with deployments using specialized hardware (Mahendran et al., 2021) and Raspberry Pi platforms with vision-language models (Baig et al., 2024). However, coordinated hardware integration (RFID, camera, audio) with quantized models and local interfaces for medication safety remains unaddressed.

This work proposes a standalone, dual-layer medication verification system leveraging the Raspberry Pi to deliver a novel assistive technology for safe and independent medication use by blind and visually impaired individuals. The primary contributions of this work are:

- A novel tri-modal verification system combining RFID scanning, real-time prescription analysis, and adaptive audio feedback for blind users' accessibility and reliability.
- Submillisecond verification pipeline with hierarchical validation of five safety axes: authenticity, timing, dosage, formulation, allergies; ensuring realtime precision.
- Fully offline, privacy-preserving edge solution that locally processes and syncs prescription data eliminating cloud reliance while ensuring secure and atomic records.
- Audio-first feedback system delivering adaptive, prioritized messages. Achieves



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 149–158, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-017

100% detection of critical medication hazards, enabling blind users to receive instant, non-visual alerts.

## 2 Related Work

Electronic adherence monitoring systems like MEMS, smart pill bottles, and ingestible sensors automatically track medication intake but often rely on cloud connectivity (Vitolins and Smith, 2022; Odhiambo et al., 2021). These systems enhance adherence through reminders and record-keeping, yet typically lack verification mechanisms beyond logging access events (Odhiambo et al., 2021; Smith and Clark, 2021). RFID-based medication management has been explored for safety and automation. The RMAIS prototype integrates an RFID reader, scale, and rotating dispenser for scheduled medicine presentation (McCall et al., 2010), while portable smart pillboxes demonstrate adherence improvements through tagged containers (Doe and Roe, 2024). Commercial solutions like ScripTalk provide audio prescription information to visually impaired users but require centralized stations and are not self-contained edge systems.

Computer vision approaches include deep-learning pill recognition with imprint detection (Heo et al., 2023), YOLO-based mobile applications with audio feedback (Dang et al., 2024), and graph-based multimodal recognition for natural scenes (Nguyen et al., 2023). However, vision-assisted solutions depend on cloud-based inference, raising privacy and latency concerns. However, most systems operate in isolation, RFID-based systems lack content verification, while vision-only approaches may misidentify pills. Some works combine modalities through ingestible RFID sensors and federated learning frameworks (Cheung and Lee, 2024), yet integration of RFID with on-device vision and audio feedback for visually impaired users remains uncommon.

Edge AI systems demonstrate feasibility for privacy-sensitive assistive devices, with successful deployments on Raspberry Pi platforms for navigation and object recognition (Mahendran et al., 2021; Baig et al., 2024; Wong et al., 2025). However, medication verification combining multimodal sensing has not been addressed.Multilingual transformers with retrieval-augmented generation show effectiveness for low-resource languages (Das et al., 2025a), though not yet applied to assistive medication systems. Future extensibility includes enhanced security through modified RSA algorithms for RFID data protection (Das et al., 2019) and multilingual neural machine translation for global deployment in diverse linguistic communities (Bala Das et al., 2023). While prior work established RFID-based dispensing, vision-driven recognition, and edge-deployed assistive systems, a critical gap remains: dual-modality (RFID + AI vision + audio) medication verification running entirely on device. Our device addresses this gap by integrating RFID tag reading, on-device Gemma3 vision analysis, and text-to-speech feedback within a portable Raspberry Pi platform for visually impaired medication management.

## 3 System Architecture

DRISHTI delivers real-time medication verification through multimodal sensing and on-device AI processing within a compact Raspberry Pi platform. Operating entirely offline, the system integrates RFID scanning, vision processing, and audio feedback for privacy-preserving medication safety.

### 3.1 System Components

The system integrates a Raspberry Pi 4 Model B (8GB RAM) with three input modalities: MFRC522 RFID reader (13.56 MHz) for contactless scanning, Pi Camera Module v3 for prescription capture, and Bluetooth/WiFi for wireless synchronization. RFID tags encode compressed medication data using a concise CSV format that embeds seven essential attributes (`med_id, dosage_schedule, form_code, expiry_date, strength, brand_name, generic_name`) within a single line. This encoding addresses the 52-byte storage limitation of standard RFID tags while achieving approximately 75% data compression compared to conventional JSON representations, with lexical pattern analysis.

The accessibility interface provides *pyttsx3* text-to-speech, tactile controls, and GPIO LEDs/buzzers. The core *MedicationClassifier* employs hierarchical validation using confidence-weighted scoring: exact matching (100%), generic equivalence (95%), therapeutic substitution (90%), and fuzzy similarity

(85%). A SQLite3 *PharmaceuticalDatabase* manages 248,000 drug entries with brand-to-generic mappings, therapeutic networks spanning thirteen drug classes, and allergy matrices. The Gemma3 4B quantized vision model processes prescription documents with 4-bit quantization for real-time edge performance, supported by *SimpleMFRC522*, *watchdog*, and *pandas/NumPy* libraries.

### 3.2 Integration and Workflow

The system supports dual-mode prescription acquisition via camera-based digitization and wireless entry (Bluetooth/Wi-Fi). Captured images are processed by the vision engine to extract structured medication data, while external devices can transmit prescriptions directly. All inputs are standardized into a unified JSON format including patient demographics, regimens, allergy profiles, and physician details. Input pathways RFID scanning, camera capture, and wireless input converge at the multi-criteria classification engine for validation against prescription profiles. The system performs prescription matching ($\tau = 85\%$), timing checks, dosage comparison ($\delta = 0.1$ mg), and safety screening with prioritized decision trees. Context-aware audio feedback is generated through a dynamic text-to-speech module, which varies tone, speed, and urgency by safety category (safe: 160 WPM, warnings: 150 WPM, urgent: 140 WPM), embedding drug name, strength, and schedule. This ensures patient-specific feedback calm confirmations, cautious warnings, urgent alerts rather than generic templates, improving clarity and trust. Processing is fully local with sub-millisecond response times, and all interactions are logged with timestamps and confidence metrics.

### 3.3 Dataset Description

The pharmaceutical knowledge base was built from a Kaggle dataset (Singh, 2023) containing over 248,000 medicines with usage, side effects, and substitutes. It provides structured attributes such as brand and generic names, therapeutic classes, dosage details, and equivalent substitutes, enabling construction of the hierarchical drug ontology for brand–generic mappings, therapeutic substitution networks, and allergy cross-reference tables. To fit the resource-constrained edge device, preprocessing removed non-essential text and reduced memory load. Drug names and substitutes were normalized, duplicates eliminated, and therapeutic equivalence pairs extracted for substitution checks. Side-effect and allergy data were converted into structured forms for real-time lookups, supporting efficient management of 248,000 entries on the Pi without compromising accuracy or response time.

## 4 Methodology

### 4.1 Design Framework

The system architecture integrates four principal modules: RFID-based medication identification, prescription data acquisition via optical character recognition and manual entry, an intelligent classification engine incorporating therapeutic-equivalence matching, and an accessibility-centric multimodal feedback generator to ensure end-to-end verification and user-friendly interaction tailored for visually impaired users.

The system is deployed on a compact, edge-computing platform that integrates essential hardware to enable robust, real-time medication verification. Key components include an MFRC522 RFID scanner operating at 13.56 MHz for tag detection, a camera module for digitizing prescription documents, and an onboard audio subsystem delivering adaptive text-to-speech prompts. A three-button tactile interface with raised indicators facilitates non-visual navigation, while Bluetooth connectivity supports wireless synchronization of prescription data. By combining RFID, camera, and Bluetooth inputs into a unified tri-modal architecture managed entirely by the local processor. The system provides redundant, flexible pathways for accurate verification tailored to visually impaired users.

The complete system workflow, depicted in Figure 1, delineates a tri-modal input architecture comprising RFID-based wave sensing, camera-driven image scanning, and manual wireless input, whose data streams are consolidated by a portable edge-computing device to generate real-time audio feedback for users with visual impairments.

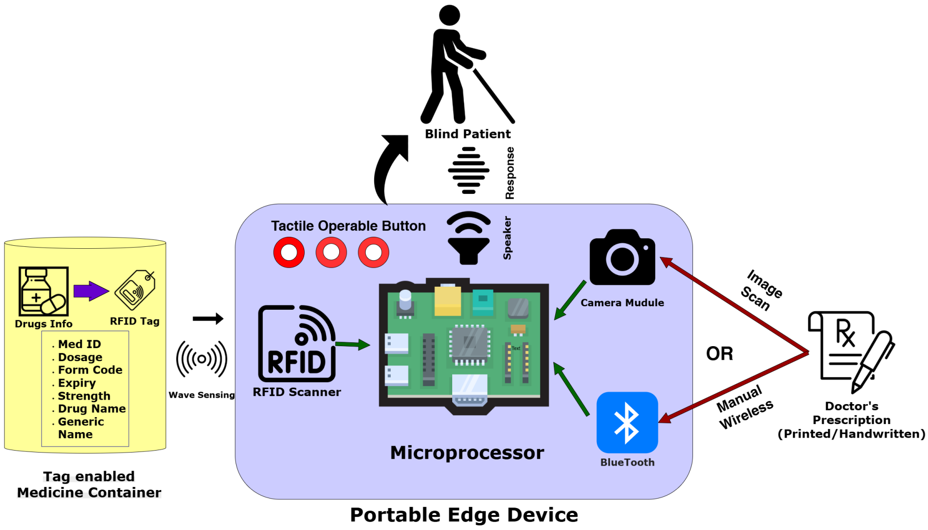



Figure 1: Workflow of DRISHTI, showing tri-modal medication verification on the edge-device using RFID scanning, camera-based prescription analysis, wireless input, and audio output capability

### 4.2 Multi-Source Drug Database and Intelligent Drug Matching

The system incorporates a comprehensive pharmaceutical knowledge base of 248,218 drug entries with hierarchical relationships including brand-to-generic mappings (25 equivalence pairs), therapeutic substitution networks spanning 13 major drug classes (ACE inhibitors, proton pump inhibitors, statins), and contraindication matrices for allergy cross-referencing and drug interactions.

The medication-matching framework employs confidence-weighted scoring through cascading similarity metrics: exact string matching (100% confidence), generic equivalence matching (95% confidence), therapeutic substitution (90% confidence), and fuzzy string matching (85% confidence), ultimately returning the highest-confidence match for prescription validation.

### 4.3 Multi-Criteria Safety Classification

DRISHTI uses a hierarchical, multi-stage validation pipeline to ensure robust, safe medication verification, as in Table 1. The five stages address prescription accuracy, dosage correctness, timing adherence, formulation compatibility, and overall safety. The system begins with *prescription verification*, cross-referencing the scanned medication against active prescriptions using exact/fuzzy string matching with a confidence threshold of $\tau = 85\%$. *Temporal validation* is then performed, checking for medication intake within flexible windows: morning (05:00–14:00), afternoon (14:00–20:00), and evening (20:00–05:00).

Subsequently, *dosage accuracy verification* ensures prescribed and scanned strengths within a precision tolerance of $\delta = 0.1$ mg. *Formulation compatibility* evaluates acceptability across alternative forms using a standardized taxonomy, and *safety screening* checks for expiration and allergies, referencing grouped categories such as penicillin, sulfa, or cephalosporin families. DRISHTI's deterministic, hierarchically prioritized decision tree classifies medications as *NOT_PRESCRIBED*, *EXPIRED*, *WRONG_STRENGTH*, *WRONG_TIMING*, or *CORRECT*. Dangerous cases trigger urgent alerts, while non-critical timing deviations prompt guidance. An accessibility-first, multimodal interface delivers real-time, context-aware feedback via offline text-to-speech and tactile controls, ensuring intuitive verification and safety across user abilities.The decision trees in DRISHTI are hand-crafted and rule-based, rather than learned from training data. This design choice was made to ensure transparency, interpretability, and auditability, which are essential in safety-critical applications. Each branch directly corresponds to medically relevant checks—such as prescription match, dosage tolerance, expiry validation, or allergy screening—ensuring predictable behavior under all conditions. While machine-learned classifiers could capture more subtle patterns, the deterministic approach minimizes false negatives and enables regulatory compliance through explainable rules.

### 4.4 Real-Time Performance

The system adopts an asynchronous, event-driven architecture that ensures continuous RFID monitoring with average response latency $< 1.0$ millisecond. Atomic file operations and write-ahead logging guarantee data consistency and thread safety during concurrent prescription updates. The unified tri-modal input architecture coordinates RFID detection (for immediate parsing of CSV/JSON medication data), real-time vision capture (triggering LLM-driven OCR for prescription labels), and maintains updated medication profiles using a local SQLite3 database, enabling reliable offline operation in resource-constrained settings.

Thread-safe mechanisms, including the *PrescriptionFileHandler* (with watchdog-based monitoring), prevent race conditions by automatically reloading modified prescription files. This maintains current, accurate medication data and detailed local logs with precise timestamps, ensuring robust state management, rapid verification, and immediate user feedback in high-frequency medication scenarios. The system provides context-aware audio feedback optimized for visually impaired users through offline text-to-speech processing. Feedback messages use differentiated tones and speech rates to convey urgency levels, with graded responses detailed in Table 2 enabling users to distinguish between safe confirmations, cautionary guidance, and critical alerts.

## 5 Evaluation and Results

### 5.1 Experimental Setup

We developed a systematic evaluation framework consisting of 149 test scenarios, which were divided into seven medication safety categories. Table 3 summarizes the distribution of these scenarios, including their respective counts and percentages. The dataset includes a balanced mix of correct and incorrect medication use cases, such as perfect matches, valid substitutes, wrong timing, dosage mismatches, form mismatches, expired medications, and drug interactions. This categorization ensured that the evaluation comprehensively addressed both routine and safety-critical situations.

The evaluation framework relied on expert-annotated ground truth labels, where each test scenario was classified as either *SAFE* or *DANGEROUS* and assigned an associated confidence score. These labels served as the reference standard for performance assessment. System predictions were compared against the expert classifications using a multi-method correctness determination approach. In cases of ambiguity, a safety-first fallback mechanism was applied to prioritize conservative decisions, ensuring that potentially dangerous scenarios were never misclassified as safe.

### 5.2 Performance Results

Our comprehensive evaluation demonstrates robust performance across all tested scenarios, as illustrated in Figure 6. The system achieved an overall accuracy of 86.57% when evaluated across 149 test scenarios, highlighting its effectiveness in verifying medication safety. Performance analysis revealed that scenarios classified as safe were correctly identified with an accuracy of 77.14%, while all safety-critical scenarios were detected with perfect accuracy (100.0%). For cases involving timing or minor safety issues, the system achieved an accuracy of 80.0%, reflecting its ability to provide appropriate warnings in non-critical situations. The mean response time for a complete verification cycle was measured at approximately 1.0 ms, confirming the system's suitability for real-time operation. Detailed per-classification performance metrics are provided in Figure 2, and the distribution of the test scenarios is summarized in Figure 3.

#### 5.2.1 Per-Classification Performance

Performance varied across different classification types, with the highest accuracy observed for safety-critical categories as shown in Table 4 and (Figure 2). The system achieved perfect accuracy (100.0%) for both *CORRECT* classifications (58/58) and *WRONG_TIMING* cases (28/28), indicating reliable detection of properly prescribed medications and correct identification of timing-related deviations. Similarly, detection of expired medications achieved 100.0% accuracy (13/13), while strength mismatches were identified with an accuracy of

| Stage | Validation Step | Description |
|---|---|---|
| 1 | Prescription Verification | Exact/therapeutic/fuzzy matching; confidence $\tau = 85\%$ |
| 2 | Temporal Validation | Morning: 05:00–14:00; Afternoon: 14:00–20:00; Evening: 20:00–05:00 |
| 3 | Dosage Accuracy | Compare prescribed vs. actual strength; tolerance $\delta = 0.1$ mg |
| 4 | Formulation Compatibility | Validate acceptable substitutes (e.g., tablet $\leftrightarrow$ capsule) |
| 5 | Safety Screening | Check expiry date and patient-specific allergy conflicts |

Table 1: Five-stage hierarchical validation framework for medication safety.

| Alert Type | Tone / Speed | Example Message |
|---|---|---|
| Safe Confirmation | Calm / 160 WPM | "This is your Lisinopril 10mg. Appropriate timing for morning dose. Safe to take." |
| Warning Alert | Cautious / 150 WPM | "This is your correct medication, but it's 2 hours early. Next dose recommended at 8 PM." |
| Danger Alert | Urgent / 140 WPM | "STOP! Wrong strength detected. You have 20mg but prescribed 10mg. Do not take." |

Table 2: Examples of adaptive audio responses with tone and speed variations.

| Scenario Category | Count (n) | Percentage (%) |
|---|---|---|
| Perfect Match Scenarios | 39 | 26.2 |
| Valid Substitute Scenarios | 30 | 20.1 |
| Wrong Timing Scenarios | 25 | 16.8 |
| Dosage Mismatch Scenarios | 20 | 13.4 |
| Form Mismatch Scenarios | 12 | 8.1 |
| Expired Medication Scenarios | 15 | 10.1 |
| Drug Interaction Scenarios | 8 | 5.4 |

Table 3: Distribution of the 149 test scenarios across seven medication safety categories.

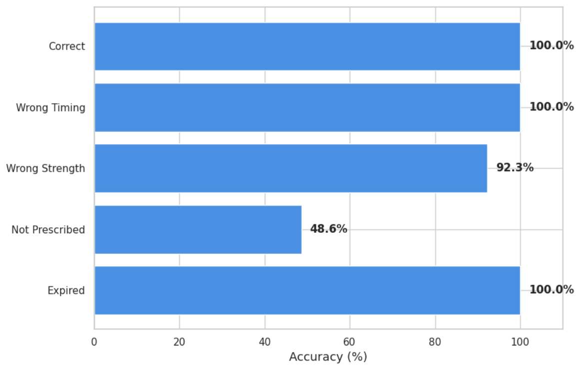



Figure 2: Classification-Specific performance with accuracy breakdown across different classification categories.

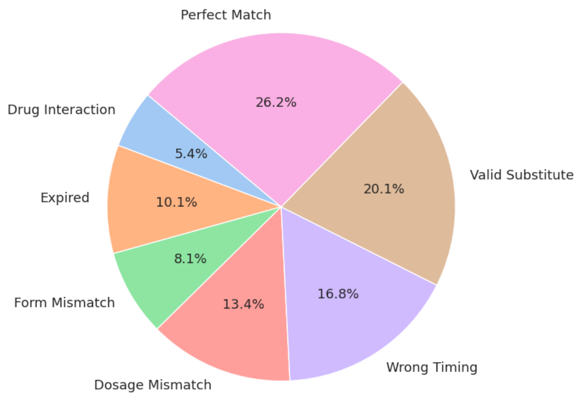



Figure 3: Test Scenario Distribution showing balanced distribution across 149 test scenarios

| Classification Type | Accuracy (%) | Cor./Tot. |
|---|---|---|
| CORRECT Classification | 100.0 | 58/58 |
| WRONG_TIMING Classification | 100.0 | 28/28 |
| EXPIRED Detection | 100.0 | 13/13 |
| WRONG_STRENGTH Detection | 92.3 | 12/13 |
| NOT_PRESCRIBED Detection | 48.6 | 18/37 |

Table 4: Classification-specific performance results across all test scenarios.

92.3% (12/13).

The lowest performance occurred for *NOT_PRESCRIBED* cases: 48.6% correct (18/37). This outcome is intentional—DRISHTI employs a conservative policy that flags any unrecognized medication for manual verification to avoid false-safe classifications. Errors primarily stemmed from (i) brand–generic mismatches (generic in the prescription vs. branded RFID not in the database) and (ii) regional formulations missing from the dataset. Mitigation will include expanded brand–generic normalization and incorporation of regional drug vocabularies.

#### 5.2.2 Response Time and Confidence Analysis

Table 5 summarizes the distribution of response times for all verification cycles. The system demonstrates exceptional processing efficiency, achieving a mean response time of approximately 1.00 ms. Notably, 98.0% of all verification cycles are completed in under 0.5 ms, while only 0.67% require between 0.5 and 1.0 ms, and 1.33% exceed 1.0 ms. These results confirm that DRISHTI delivers ultra-fast, real-time performance with no perceptible delay during user interaction, a critical factor for assistive devices deployed on edge platforms.

| Response Time (ms) | Scenarios (n) | Percentage (%) |
|---|---|---|
| 0 – 0.5 | 146 | 98.0 |
| 0.5 – 1.0 | 1 | 0.67 |
| > 1.0 | 2 | 1.33 |
| **Mean Response Time** | – | **1.00 ms** |

Table 5: Distribution of response times for all verification cycles.

The confidence-accuracy correlation analysis further highlights the robustness of the classification engine. High-confidence predictions (95–100%) for *CORRECT* scenarios consistently achieve 100% accuracy, while *NOT_PRESCRIBED* classifications reach 48.6% accuracy at similar confidence levels. For safety-critical cases, the system achieves 100% accuracy for expired medications when predictions are made at full confidence, and 92.3% accuracy for wrong-strength detections when predictions are made with 90% confidence. These findings demonstrate that the classifier's confidence scores reliably reflect prediction accuracy, allowing the system to adopt a safety-first strategy by flagging uncertain cases for user verification rather than risking false safe classifications.

#### 5.2.3 Vision Model Performance

The Gemma3 4B 4-bit quantized multimodal model achieves 78.4% overall prescription document analysis accuracy suitable for real-time applications. Printed prescriptions significantly outperform handwritten documents (85.2% vs 67.8% accuracy), with electronic prescriptions achieving the highest accuracy at 92.1% as shown in Figure 4. Document quality directly impacts performance, ranging from 92.1% for electronic documents to 58.9% for poor handwritten prescriptions. Information extraction accuracy varies by data type: medication names (88.5%), dosage (82.1%), schedule/frequency (76.3%), and special instructions (71.8%).

### 5.3 Safety Performance Analysis

As summarized in Table 7 and illustrated in Figure 6, DRISHTI meets its safety-first objective with 100% detection across all dangerous scenarios and zero false negatives; expired medications were detected at 100% (13/13) and incorrect strength at 92.3% (12/13). In operational tasks, correct medication identification achieved 100% (58/58) and timing validation reached 100% (28/28), demonstrating robust day-to-day reliability. Performance on *NOT_PRESCRIBED* cases shows 48.6% accuracy by design, reflecting a conservative policy that flags uncertain or unrecognized medications for manual review to avoid false-safe outcomes. The safety-first treatment of unknowns is visualized in Figure 5, underscoring that no unsafe instance is misclassified as safe.

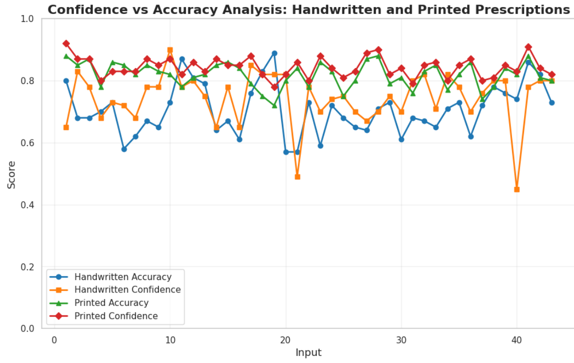



Figure 4: Handwritten vs printed prescription performance analysis

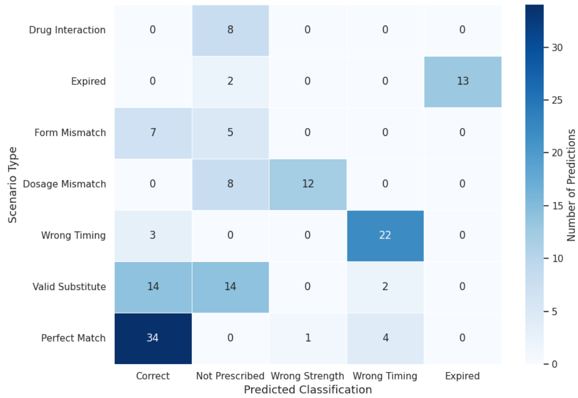



Figure 5: Confusion Matrix Analysis

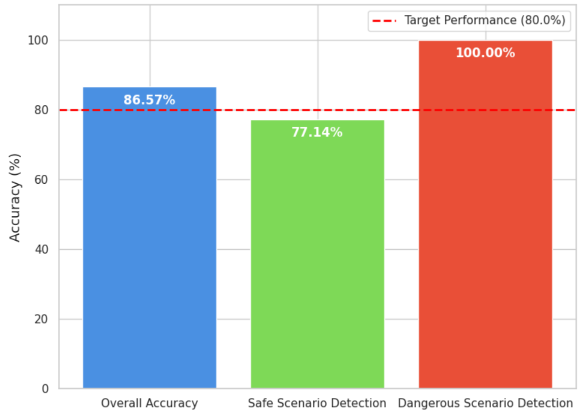



Figure 6: System Performance data

| System | Input Types | AI Processing | Awareness | Intelligent Matching | Smart Feedback |
|---|---|---|---|---|---|
| ScripTalk (En-Vision America, 2024) | Audio only | None | Limited | Basic | Static |
| YOLO-OCR (Dang et al., 2024) | Vision only | Basic | None | Simple | Static |
| RMAIS (McCall et al., 2010) | RFID only | None | None | Direct | None |
| Smart Pillboxes (Doe and Roe, 2024) | RFID only | None | Limited | Direct | Basic |
| **DRISHTI** | **Tri-modal** | **Advanced** | **Full** | **Multi-level** | **Adaptive** |

Table 6: Intelligence and AI capability comparison across medication assistance systems

| Safety Level | Correct/Total | Accuracy (%) |
|---|---|---|
| Safe Scenarios | 54/70 | 77.1 |
| Dangerous Scenarios | 43/43 | 100.0 |

Table 7: Performance across safety-critical and operational categories.

### 5.4 Comparison with Existing Solutions

To contextualize DRISHTI's capabilities, Table 6 contrasts DRISHTI with traditional medication aids that focus on pill identification or static audio (e.g., YOLO-OCR imprint reading, ScripTalk label playback, RFID-only adherence logs). DRISHTI delivers broader, intelligent management: 86.57% overall accuracy with 100% detection of safety-critical cases (expired drugs, dosage mismatches, interaction risks). Running fully offline on edge hardware preserves privacy, usability, and reliability. Moving beyond lookup, DRISHTI enables context-aware decisions via the Gemma3 4B model, multi-level matching (exact, generic, therapeutic, fuzzy), patient history and timing awareness, and adaptive urgency-based audio feedback. This comprehensive AI integration positions DRISHTI as a first, safety-first, truly intelligent assistive medication system.

## 6 Conclusion

DRISHTI is an assistive system that enhances medication safety for visually impaired users by integrating RFID identification, AI-driven visual recognition, and real-time audio interaction. Running entirely offline on low-cost edge hardware, it ensures multimodal verification with strong privacy and no cloud dependency. Evaluation across 149 scenarios shows 86.57% overall accuracy and 100% detection of safety-critical events (expired drugs, dosage mismatches, interaction risks), supporting home and institutional use. Real-time performance (<1 ms) with urgency-aware feedback enables daily integration, while the tri-modal architecture ensures fault tolerance and autonomy through voice prompts.

Future work targets multilingual scalability by integrating OCR for non-English scripts (Indic, Bangla, Arabic), expanding brand–generic mappings, and adopting multilingual text-to-speech, alongside fine-tuning vision models for diverse scripts. Prior works on Multilingual Neural Machine Translation (MNMT) for Indic-to-Indic languages (Bala Das et al., 2024) provide a foundation, while DRISHTI-Plus may leverage MNMT for multilingual dialogue and audio description (Bala Das et al., 2023). Integration with secure mobile/cloud dashboards could enhance monitoring with federated or edge-assisted learning approaches(Paul et al., 2025). To extend device capability, error analysis of language translations using the MQM framework (Das et al., 2025b) is included. Collectively, DRISHTI demonstrates real-world readiness and a clear pathway toward accessible, intelligent, and inclusive medication management for underserved populations.

## 7 Ethics Statement and Limitations

DRISHTI is designed to run fully offline, ensuring user control of sensitive data. . A conservative confidence threshold minimizes safety risks, and drug information comes from public, anonymized sources to reduce bias, although cultural and linguistic diversity remain challenges. The system is meant to assist, not replace, professional medical care. DRISHTI performs better on printed than handwritten prescriptions and is currently limited to English and Western pharmaceutical data, restricting usability in multilingual regions. Conservative detection of non-prescribed drugs increases false alerts, and hardware limitations prevent real-time updates. Clinical validation is pending, and the audio-tactile interface is insufficient for users with multiple impairments.

## References


Mirza Samad Ahmed Baig, Syeda Anshrah Gillani, Shahid Munir Shah, Mahmoud Aljawarneh, Abdul Akbar Khan, and Muhammad Hamzah Siddiqui. 2024. Ai-based wearable vision assistance system for the visually impaired: Integrating real-time object recognition and contextual understanding using large vision-language models.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2023. Improving multilingual neural machine translation system for indic languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(6).

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, and Asif Ekbal. 2024. Multilingual neural machine translation for indic to indic languages. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 23(5):1–32.

Rupert R. A. Bourne, Jaimie Adelson, Seth Flaxman, Paul Briant, Michele Bottone, Theo Vos, Kovin Naidoo, Tasanee Braithwaite, Maria Cicinelli, Jost Jonas, Hans Limburg, Serge Resnikoff, Alex Silvester, Vinay Nangia, and Hugh R. Taylor. 2020. Global prevalence of blindness and distance and near vision impairment in 2020: progress towards the vision 2020 targets and what the future holds. *Investigative Ophthalmology & Visual Science*, 61(7):2317–2317.

A. Cheung and B. Lee. 2024. Rfid based health adherence medicine case using fair federated learning. *arXiv*.

Bo Dang, Wenchao Zhao, Yufeng Li, Danqing Ma, Qixuan Yu, and Elly Yijun Zhu. 2024. Real-time pill identification for the visually impaired using deep learning.

Sajeeb Das, Srijit Paul, Akib Jayed Islam, Sultanus Salehin, and Prasun Datta. 2025a. Development of a multilingual climate fact-checking system with unified dataset for low-resource indic languages. In *Proceedings of the 16th International IEEE Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IIT Indore, Madhya Pradesh, India. IEEE Electronics Packaging Society and All India Council for Technical Education (AICTE). Paper ID: 7300.

Sudhansu Bala Das, Samujjal Choudhury, Tapas Kumar Mishra, and Bidyut Kr Patra. 2025b. Investigating the effect of backtranslation for indic languages. In *Proceedings of the First Workshop on Natural Language Processing for Indo-Aryan and Dravidian Languages*, pages 152–165.

Sudhansu Bala Das, Sugyan Kumar Mishra, and Anup Kumar Sahu. 2019. A new modified version of standard rsa cryptography algorithm. In *Smart Computing Paradigms: New Progresses and Challenges: Proceedings of ICACNI 2018, Volume 2*, pages 281–287. Springer.

J. Doe and A. Roe. 2024. Development of a smart pillbox with passive rfid to support medication adherence. *CMS Conferences*.

En-Vision America. 2024. Scriptalk station: Audio prescription reader for the visually impaired. https://www.envisionamerica.com/scripttalk. Accessed: 2024-12-01.

Shibakali Gupta, Sumana Das, and Surabhi Pal. 2023. Medication identification and assistive system for the visually impaired: Vismed. *International Journal of Computer Applications*, 186(38):30–33.

J. Heo, Y. Kang, S. Lee, D. Jeong, and K. Kim. 2023. An accurate deep learning–based system for automatic pill identification. *JMIR*.

Jagadish K Mahendran, Daniel T Barry, Anita K Nivedha, and Suchendra M Bhandarkar. 2021. Computer vision-based assistance system for the visually impaired using mobile edge artificial intelligence. In *CVPR Workshops (MAI)*.

C. McCall, B. Maynes, C. Z. Zou, and N. J. Zhang. 2010. Rmais: Rfid-based medication adherence intelligence system. In *EMBC*.

Vidula V Meshram, Kailas R Patil, Vishal A Meshram, and Shripad Bhatlawande. 2021. Smartmedbox: A smart medicine box for visually impaired people using iot and computer vision. *RIA*.

Anh Duy Nguyen, Huy Hieu Pham, et al. 2023. High accurate and explainable multi-pill detection framework with graph neural network. *arXiv*.

C. Odhiambo, P. Wright, C. Corbett, and H. Valafar. 2021. Medsensor: Medication adherence monitoring using neural networks on smartwatch accelerometer data. *arXiv*.

Srijit Paul, Sajeeb Das, Akib Jayed Islam, Sultanus Salehin, and Prasun Datta. 2025. Federated learning for privacy-preserving gait recognition on edge devices using imu data. In *Proceedings of the 16th International IEEE Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IIT Indore, Madhya Pradesh, India. IEEE Electronics Packaging Society and All India Council for Technical Education (AICTE).

Shudhanshu Singh. 2023. 250k medicines – usage, side effects and substitutes. https://www.kaggle.com/datasets/shudhanshusingh/250k-medicines-usage-side-effects-and-substitutes. Accessed: 2025-07-25.

J. A. Smith and R. Clark. 2021. A review on emerging smart technological innovations in healthcare: Medication adherence. *Smart Health Review*.

M. Z. Vitolins and C. P. Smith. 2022. Technologies for medication adherence monitoring and technology assessment. *JMIR mHealth and uHealth*, 10(3):e35157.

X. Wong et al. 2025. A hybrid object detection approach for visually impaired persons using attention-driven deep learning. *Scientific Reports*.

Fateme Zare, Paniz Sedighi, and Mehdi Delrobaei. 2023. A wearable rfid-based navigation system for the visually impaired. *arXiv*.

# What Language(s) Does Aya-23 Think In? How Multilinguality Affects Internal Language Representations

**Katharina Trinley[1,*], Toshiki Nakai[1,*], Tatiana Anikina[2], Tanja Baeumel[2]**
[1]Saarland University
[2]German Research Center for Artificial Intelligence (DFKI)
{katr00001, tona00002}@stud.uni-saarland.de

## Abstract

Large language models (LLMs) excel at multilingual tasks, yet their internal language processing remains poorly understood. We analyze how Aya-23-8B, a decoder-only LLM trained on balanced multilingual data, handles code-mixed, cloze, and translation tasks compared to predominantly monolingual models like Llama 3 and Chinese-LLaMA-2. Using logit lens and neuron specialization analyses, we find: (1) Aya-23 activates typologically related language representations during translation, unlike English-centric models that rely on a single pivot language; (2) code-mixed neuron activation patterns vary with mixing rates and are shaped more by the base language than the mixed-in one; and (3) Aya-23's language-specific neurons for code-mixed inputs concentrate in final layers, diverging from prior findings on decoder-only models. Neuron overlap analysis further shows that script similarity and typological relations impact processing across model types. These findings reveal how multilingual training shapes LLM internals and inform future cross-lingual transfer research. The code and dataset are publicly available[1].

## 1 Introduction

Large language models (LLMs) excel in multilingual tasks (Srivastava et al., 2022; Bang et al., 2023; Gurgurov et al., 2025b), but their internal handling of multiple languages remains underexplored (Kaddour et al., 2023). While methods like logit lens (Wendler et al., 2024; Schut et al., 2025) and neuron specialization (Tang et al., 2024; Kojima et al., 2024; Tan et al., 2024) have been applied, prior work mainly targets English-centered models on monolingual tasks (e.g., cloze or repetition tasks), rather than balanced multilingual architectures and their processing of code-mixed texts.

Multilingual models often default to English during intermediate processing, as described by the Multilingual Workflow (MWork) hypothesis (Zhao et al., 2024), which suggests LLMs convert non-English inputs into English internally before generating outputs. Supporting this, studies on reasoning language models (RLMs) (Wang et al., 2025) find reliance on internal "pivot" languages or scripts, even with other input languages. However, it remains unclear if this preference is unique to RLMs or a general pattern in all multilingual LLMs. Therefore, we ask:

> **H1:** How do balanced multilingual models process translation tasks – do they activate multiple languages simultaneously, unlike English-centric models that rely on a single pivot language?

Neuron-level analyses have identified language-specific patterns (Kojima et al., 2024; Tang et al., 2024), but these studies predominantly examine English-based models, leaving open whether multilingual training leads to fundamentally different internal processing mechanisms. While LLMs' language capabilities are tied to specific neuron subsets, particularly in early and late layers (Kojima et al., 2024; Tang et al., 2024), these patterns may not apply to models trained on diverse multilingual data (Zhong et al., 2024a; Schut et al., 2025). We thus investigate the following hypotheses:

> **H2:** What patterns of neuron sharing of language specific neurons emerge in balanced multilingual models, and do these align more strongly with language similarity compared to predominantly monolingual models?

> **H3:** Where do language-specific neurons concentrate in multilingual architectures – do they cluster predominantly in final layers, contrary to prior findings showing distribution across early and late layers in decoder-only models?

In real-world contexts, speakers often mix languages within a single utterance, requiring models to dynamically switch between language-specific representations. Code mixing (CM) provides a

*Equal contribution
[1]https://github.com/KatharinaTrinley/multilingual-internal-representations



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 159–171, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-018

valuable lens for studying multilingual processing in language models (Xie et al., 2025), and while multilingual LLMs perform well on some tasks, they still struggle with code-switched text (Gundapu and Mamidi, 2020). The development of more balanced multilingual models, such as Aya-23 (Aryabumi et al., 2024), offers an opportunity to examine how different training approaches affect internal language representations, especially when handling the linguistic complexity of code-mixed inputs. Thus, we ask:

> **H4:** How does the processing of code-mixed inputs vary based on language pair characteristics and models?

To address these questions, we perform a neuron-level comparison of a balanced multilingual model (Aya-23-8B), a predominantly English-trained model (Llama 3.1-8B), and a language-specialized model (Chinese-LLaMA-2-7B). Specifically, we:

**I. Analyze internal language representations** across 13-language translation tasks using logit lens to test **H1**, checking whether Aya-23 activates multiple languages simultaneously, unlike English-pivot processing in mostly monolingual models.

**II. Create a controlled code-mixed dataset** with varying mixing ratios across 10 typologically diverse pairs ($\{$fr, zh$\} \times \{$en, es, it, ja, ko$\}$) and use neuron specialization (activation frequency (Tan et al., 2024)) to investigate **H2** and **H4**, exploring how script similarity and language relationships affect neuron sharing across models.

**III. Examine layer-wise distribution of language-specific neurons** via activation strength (Kojima et al., 2024) to test **H3**, determining whether balanced multilingual training concentrates language-specific neurons mainly in final layers, contrasting prior findings of early-and-late layer distributions in decoder-only models.

## 2 Methodology

We investigate the internal language representations in multilingual decoder-only LLMs through complementary experimental approaches: logit lens analysis (Section 2.3) and neuron specialization analysis (Section 2.4). Each methodology offers unique insights into how models process information across languages.

### 2.1 Models

We evaluate three models with varying multilingual focus. **Aya-23-8B** by Cohere AI is an open-source decoder-only model instruction fine-tuned on 23 languages—including ar, zh (simplified & traditional), en, fr, it, ja, ko, and more—using a two-stage process: pretraining on a balanced multilingual corpus (not public) and multilingual instruction fine-tuning (Aryabumi et al., 2024). **Llama 3.1-8B** supports 8 languages (en, fr, de, hi, it, pt, es, th) but was mainly trained on English data (ca. 8% multilingual tokens) and retains English-centric processing patterns, serving as a baseline for predominantly English-trained models (Grattafiori et al., 2024; Wendler et al., 2024). **Chinese-LLaMA-2-7B** is a Mandarin-adapted LLaMA-2 variant with an expanded tokenizer (+20,000 tokens), pretrained on large Chinese corpora using parameter-efficient fine-tuning (LoRA(Hu et al., 2021)) and instruction-tuned on millions of Chinese instruction-response pairs, enabling strong Chinese performance at low computational cost (Cui et al., 2023a,b; Hu et al., 2021).

### 2.2 Datasets

In this work, we focus on two primary datasets: the Dumas dataset (Dumas et al., 2024) for logit lens experiments and introduce a new code-mixed dataset that will be publicly released.

**Dumas Dataset** For logit lens experiments, we use the dataset from Dumas et al. (2024), which includes word translation and cloze tasks in 13 languages (de, en, es, et, fi, fr, hi, it, ja, ko, nl, ru, zh). It minimizes token overlap between languages while maintaining semantic consistency. Note that model support varies: Aya-23-8B lacks et and fi; Llama 3.1-8B excludes et, fi, ja, ko, nl, ru, and zh; Chinese-LLaMA-2-7B supports only zh and has limited en capabilities, lacking official support for the other 11 languages. Each prompt consists of randomly selected 5-shot word translation examples followed by a final query word. For instance, an English-to-Chinese task may appear as:

*English: "computer"* → 中文: 电脑
*English: "ant"* → 中文: 蚂蚁
*English: "cloud"* → 中文: 云
*English: "heart"* → 中文: 心脏
*English: "knife"* → 中文: 刀子
*English: "book"* → 中文: __

The task is to predict the correct translation of the final word. Synonyms for the target word are included across all supported languages.

**Code-mixed Dataset** To study how models process mixed-language inputs, we construct a code-mixed dataset derived from the WMT24++ parallel corpus (Deutsch et al., 2025), containing 998 sentence pairs across 55 languages. We focus on a subsection of 7 languages and take fr and zh as base languages, each mixed with five partner languages (en, es, it, ja, and ko) resulting in ten language pairs. These combinations span a wide typological and script range, including closely related Romance/Indo-European languages (fr/es, fr/it, fr/en), typologically distinct but historically linked pairs (zh/ja, zh/ko), and diverse scripts: Latin (en, fr, es, it), Simplified Chinese (zh), Kanji/Kana (ja), and Hangul (ko).

We generate code-mixed sentences using a three-step rule-based method (Figure 1) with controlled mixing ratios of 25%, 50%, and 75%.

We tokenize Latin script using whitespace and Han script with the Jieba library (Junyi, 2012). Although this may yield ungrammatical outputs, it ensures consistent mixing ratios critical for controlled experiments. To address limited dictionary coverage in prior work (Conneau et al., 2017), we create comprehensive bilingual dictionaries via Google Translate for all WMT24++ words, ensuring equal vocabulary coverage across language pairs. However, lacking word sense disambiguation, polysemous words are translated identically regardless of context, possibly causing meaning mismatches.

To evaluate translation accuracy, we manually assessed word-level translation quality in code-mixed data, focusing on semantic mistranslations rather than grammatical errors common in code-mixing. From 50% mixing datasets, we sampled 10 sentences per language pair (246–399 words) and found translation error rates of fr-en 4.76% , fr-es 4.78% , zh-en 4.87% , and zh-es 8.94% , with higher errors for zh-es due to greater linguistic distance and weaker model performance.

To compare code-mixed and monolingual processing, we include corresponding monolingual datasets from WMT24++ (fr, es, it, ja, and ko) as baselines. All code-mixed pairs were evaluated on translation tasks directed from code-mixed input to en (i.e., Chinese-Spanish code-mixed input to en). We do not evaluate the reverse direction, as enforcing controlled code-mixing in model-generated outputs is challenging.

To further examine model behavior, we analyze neuron activation patterns (Section 2.4) across code-mixed inputs for Aya-23-8B, LLaMA 3.1-8B, and Chinese-LLaMA-2-7B, testing whether code-mixed processing differs by language pair and model architecture (**H4**).

## 2.3 Logit Lens

Logit lens (Nostalgebraist, 2020) interprets transformer hidden states by projecting intermediate representations into vocabulary space. At each layer $\ell$, the model produces a hidden state $h_\ell \in \mathbb{R}^d$, which is mapped to logits using the unembedding matrix $U \in \mathbb{R}^{|V|\times d}$: $\text{logits}_\ell = Uh_\ell$.

These logits approximate the model's predictions at layer $\ell$. Following Nostalgebraist (2020), we use the residual stream before layer normalization to better align with the final outputs. Building on prior multilingual analyses (Wendler et al., 2024; Zhong et al., 2024b; Saji et al., 2025), we apply the logit lens at each layer, extract token probabilities via softmax, and sum over synonyms in 13 languages using the dataset from Dumas et al. (2024) (see Section 2.2). To reduce false matches, we apply a 0.1 threshold. This approach allows us to track the emergence of language-specific signals across layers and test **H1**.

## 2.4 Neuron Specialization

Neuron specialization refers to individual neurons within language models developing preferences for processing specific types of input, such as particular languages.

**Tan et al.'s Approach** Tan et al. (2024)'s method identifies language-specific neurons by measuring how frequently they activate when processing different languages. Following Tan et al. (2024), we identify language-specific neurons via binary ReLU activations in FFNs across WMT24++ and code-mixed data.

For task $t$ with validation set $D_t$, each sample $x_i$ has activation $\mathbf{a}_i^t$. Summing gives $\mathbf{a}^t = \sum_{x_i \in D_t} \mathbf{a}_i^t$. Specialized neurons $S_k^t$ are the top activations satisfying $\sum_{i \in S_k^t} \mathbf{a}^t(i) \geq k \sum_i \mathbf{a}^t(i)$. Neuron overlap is measured by $\text{IoU}(S^i, S^j) = \frac{|S^i \cap S^j|}{|S^i \cup S^j|}$. Using $k = 90\%$ per Tan et al. (2024), we identify neurons covering most activations per language and plot IoU matrices to expose cross-linguistic patterns. Unlike Tan et al. (2024), we exclude neurons shared by all languages to isolate language-specific neurons. This tests **H2**.

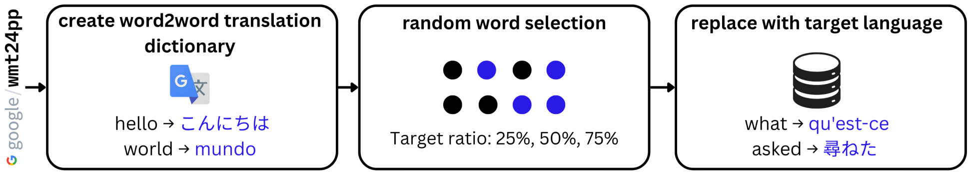



Figure 1: Our code-mixed dataset creation pipeline. Starting with parallel sentences from WMT24++, we create comprehensive bilingual dictionaries using Google Translate for all vocabulary. For each sentence, we randomly select words based on the target mixing ratio (25%, 50%, or 75%) and replace them with their translations in the partner language. For example, from the English source "The World Bank hopes to spread that message," we generate the code-mixed Chinese output "World bank希望传播这一理念" (50%).

**Kojima et al.'s Approach** Kojima et al. (2024) identified language-specific neurons in multilingual models, concentrated in early and late layers with minimal cross-language sharing. Their approach identifies neurons that discriminate between target language content and other languages by measuring activation strength.

We extend this to code-mixing neurons in Aya-23-8B's MLP layers. For each code-mixed pair $l_t$, texts are labeled positive ($b_i = 1$) or negative ($b_i = 0$). For neuron $m$ and text $x_i = \{w_{i,1}, \ldots, w_{i,T}\}$, activations $\{z_{m,i,1}, \ldots, z_{m,i,T}\}$ are averaged as $z_{m,i} = f(z_{m,i,1}, \ldots, z_{m,i,T})$ (excluding padding). We compute Average Precision $AP_m = AP(z_m, b) \in [0, 1]$ to classify neurons into top-$k$ (high), medium-$k$ (none), and bottom-$k$ (negative correlation). Applied to fr and zh code-mixed with en, it, es, jp, ko (10 pairs), this tests **H3** and **H4**.

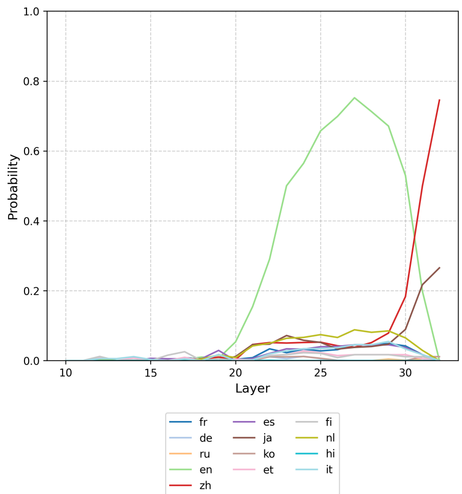



Figure 2: Logit lens language probabilities for English-to-Chinese translation in Aya-23-8B reveal activation of an increased number of languages in mid-to-late layers, with English being dominant.

## 3 Results and Discussion

### 3.1 Logit Lens Analysis

To test if balanced multilingual training affects internal processing (**H1**), we applied logit lens analysis (Wendler et al., 2024) to Aya-23-8B (balanced), LLaMA 3.1-8B (English-dominant), and Chinese-LLaMA-2-7B (Chinese-specialized).

Using Dumas et al. (2024)'s dataset, we tracked language-specific token probabilities across layers during translation. From 54 tasks, we computed AUCs for each language probability curve and used Mann-Whitney U tests with Bonferroni correction to compare: (1) model effects – whether Aya shows more diverse language representations than LLaMA ($p < 0.05/(13 \times 3) = 0.0013$), and (2) task effects – whether input vs. output languages differ in internal processing ($p < 0.05/(13 \times 3 \times 2) = 0.0006$).

Aya-23-8B demonstrates multilingual processing with cross-linguistic activation. During English-to-Chinese translation (Figure 2), Aya activates multiple languages in intermediate-to-late layers (20–27), including Japanese tokens despite Japanese being neither source nor target. This suggests Aya leverages typological relationships rather than relying solely on English as a pivot.

Llama 3.1-8B follows English-centric processing. In contrast (Figure 3), Llama demonstrates the English-dominated pattern established by Wendler et al. (2024), with English maintaining highest activation across all layers until final output generation. Chinese activates only in final layers, aligning with the "English-ization" process (Zhao et al., 2024).

Chinese-LLaMA-2-7B exhibits Chinese-dominant processing. This model shows Chinese representations dominating across most layers even for English-to-Chinese translation (Figure 4), with English activation decreasing in final layers while Japanese remains stable, reflecting its specialized training.

Our statistical analysis across all 54 translation

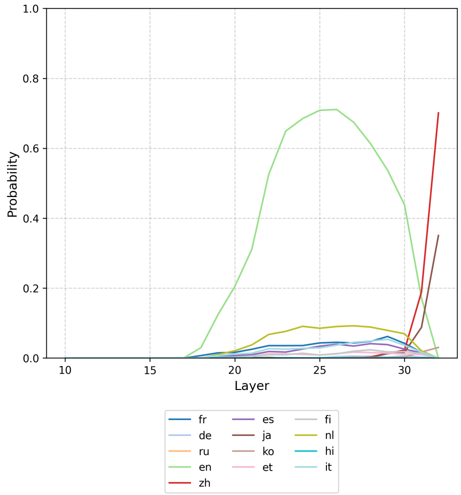



Figure 3: Logit lens language probabilities for English-to-Chinese translation in Llama 3.1-8B show dominant English representations across most layers with few other languages showing significant activation.

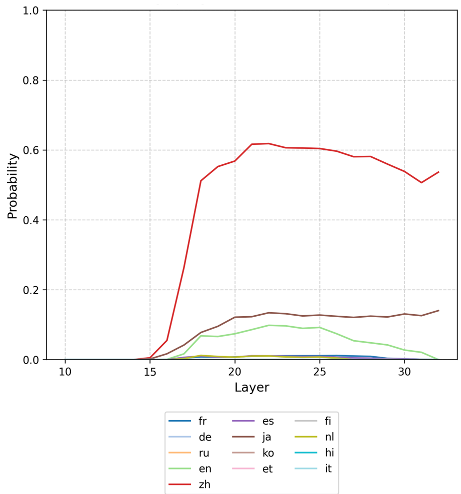



Figure 4: Logit lens language probabilities for English-to-Chinese translation in Chinese-LLaMA-2-7B show strong dominance of Chinese representations across most layers.

tasks provides quantitative support for **H1**: Aya demonstrates significantly different language activation patterns compared to both Llama (8/13 languages with $p < 0.0013$: de, ru, zh, es, ja, ko, it) and Chinese-LLaMA (8/13 languages including en, zh, es, ja, ko, it). Critically, output languages influence internal representations more strongly than input languages across all models, when analyzing task composition effects, output language presence produces significant changes in 12/13 languages compared to only 7/13 for input languages.

This analysis partially supports our hypothesis that Aya-23 incorporates multiple languages in in-

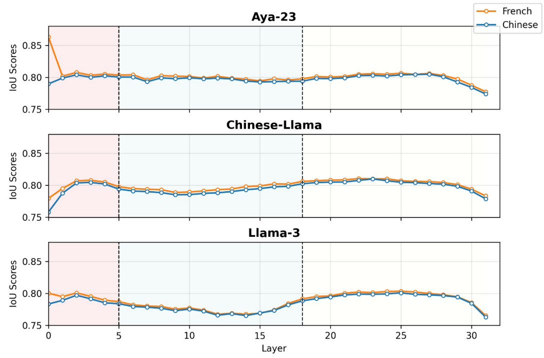



Figure 5: Three-phase neuron clustering patterns across transformer layers. French-based (orange) and Chinese-based (blue) code-mixed language pairs show distinct IoU overlap patterns in Aya-23, Chinese-LLaMA, and Llama-3.1. All models exhibit consistent French processing advantages.

ternal processing, rather than relying solely on English. However, English still shows significantly higher activation probabilities, necessitating careful interpretation of these multilingual patterns. The statistical evidence highlights that both task language and model training paradigm significantly shape internal processing strategies, with task language particularly influencing language-specific activation probabilities.

### 3.2 Neuron Specialization Analysis

**Activation *Frequency* Experiments** Following Tan et al. (2024), we conducted neuron activation frequency experiments to examine how balanced multilingual training influences language-specific processing mechanisms (**H2**, **H4**).

To investigate base-language dependencies systematically, we conducted statistical analysis comparing French-based and Chinese-based code-mixed language pairs across all 32 transformer layers (see Figure 5). For each layer, we computed IoU overlap values within French-based pairs (105 combinations from 15 tasks) and within Chinese-based pairs (105 combinations from 15 tasks), yielding two distributions of IoU scores per layer. We applied the Wilcoxon signed-rank test to assess whether French-based pairs show significantly different neuron clustering patterns than Chinese-based pairs, using this non-parametric paired test since we're comparing corresponding layers between the two language groups.

French-based code-mixed inputs demonstrate significantly higher neuron clustering than Chinese-

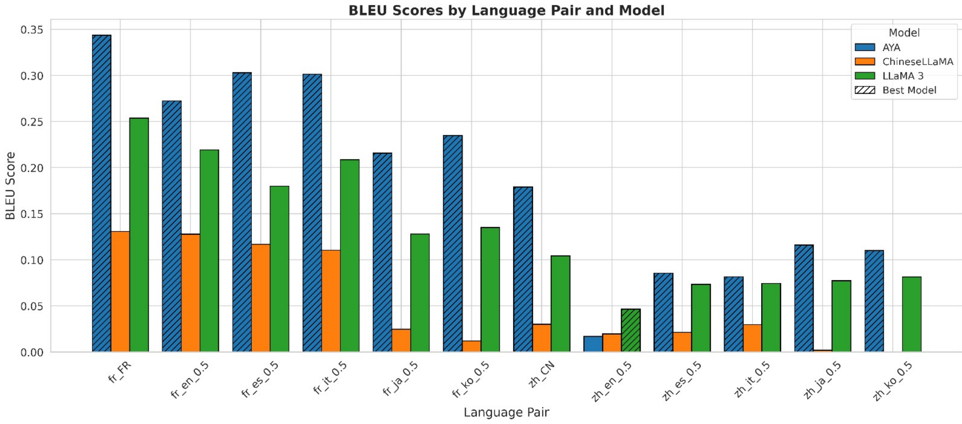



Figure 6: Translation qualities on code-mixed datasets using Aya23-8B, LLaMA 3.1-8B, and Chinese LLaMA, presented in BLEU.

based inputs across all three models. Wilcoxon signed-rank tests reveal strong statistical significance: Aya-23 ($p = 4.66 \times 10^{-10}$, mean difference = +0.0050), Chinese-LLaMA ($p = 4.66 \times 10^{-10}$, mean difference = +0.0041), and Llama-3 ($p = 9.31 \times 10^{-10}$, mean difference = +0.0029). This French advantage persists even in Chinese-LLaMA, a model specifically adapted for Chinese processing.

Our findings contradict **H2**, as neuron sharing patterns do not align with expected base-language training effects. Instead, they reveal a universal French processing advantage that transcends model architecture and training paradigm ($p < 10^{-9}$ across all models). This pattern strongly supports **H4** – that code-mixed processing varies systematically with language pair characteristics – and indicates that factors beyond training data composition, potentially including script characteristics or tokenization efficiency, drive neuron activation patterns in multilingual models.

**Translation Performance on Code-Mixed Inputs** Figure 6 presents BLEU scores for all three models on monolingual and code-mixed datasets. Aya-23-8B consistently outperforms the others, with a clear advantage on fr-based code-mixed inputs. All models show better performance on Latin-script pairs (fr-en, fr-es, fr-it) than on cross-script ones (fr-ja, fr-ko). For zh code-mixing, Aya-23-8B and Llama 3.1-8B perform better on zh-ja and zh-ko than on zh-en, zh-fr, and zh-it, suggesting that shared vocabulary and typological features help transfer despite script differences. In contrast, Chinese-LLaMA-2-7B performs poorly across all code-mixed inputs, regardless of typological similarity.

Performance generally degrades as code-mixing rate increases across all models, likely reflecting limitations of our rule-based word-to-word translation approach. However, Aya-23-8B shows greater resilience to this degradation, supporting our finding that balanced multilingual training improves robustness to code-mixing.

**Activation *Strength* Experiments** To address H4, we followed Kojima et al. (2024)'s methodology by processing both monolingual and code-mixed texts and capturing neuron activations at the MLP layers. Our findings for Aya reveal an interesting divergence from previous work on decoder-only model. While Kojima et al. (2024) found language-specific neurons (both top-k and bottom-k) concentrated in first and last layers of other decoder-only models, Aya-23-8B exhibits a different pattern when processing code-mixed input: top-k language-specific neurons appear predominantly in final layers (27-31), with a pronounced spike in layer 31 across all language pairs (see Figure 7). This pattern confirms our hypothesis **H3**.

This pattern only partially aligns with Tang et al. (2024), who observed a skewed "U"-shaped distribution, with language processing concentrated in both early and late layers. In contrast, it supports the findings of Mondal et al. (2025), who reported that language-specific neurons in modern LLMs are primarily concentrated in later layers. Our results suggest that Aya-23-8B's balanced multilingual training may promote a shift toward language-specific processing concentrated at the generation stage, diverging from the more distributed patterns

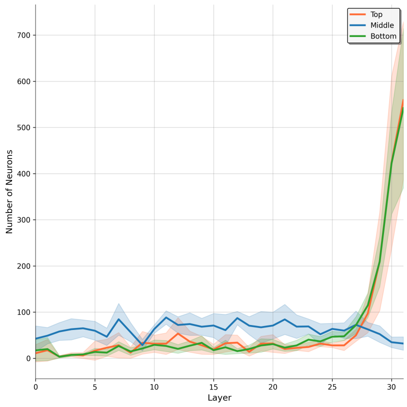



Figure 7: Layer-wise distribution of $k = 1000$ language-specific neurons in Aya-23-8B for code-mixed processing across all CM language pairs in Aya-23-8B.

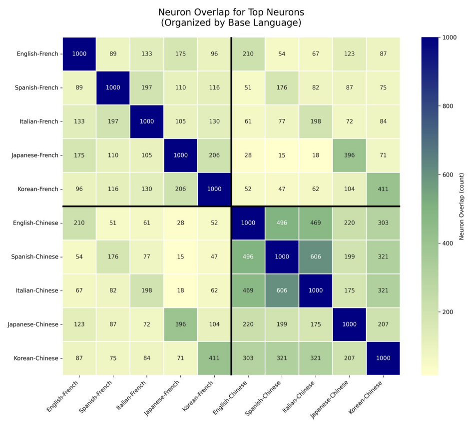



Figure 8: The number of overlapping language-specific neurons between code-mixing language pairs in Aya-23-8B.

seen in predominantly monolingual models.

This pattern remains consistent across all language pairs. Bottom-k neurons ("anti-correlated" neurons) similarly concentrate in final layers, while medium-k neurons distribute more evenly across early (0-5) and middle (10-20) layers.

This distinctive concentration pattern may stem from Aya's explicitly balanced multilingual training, resulting in an internal structure different from the predominantly monolingual models studied by Kojima et al. (2024). The pronounced spike of language-specific neurons in the final layer likely reflects Aya-23-8B's processing strategy for code-mixed inputs: earlier layers handle distributed multilingual representations for understanding, while the concentration at the generation stage resolves which language to output in mixed-language contexts.

Our analysis of neuron overlap shows that the base language influences neuron sharing more than the secondary mixed-in language. Chinese-based pairs consistently exhibit higher neuron overlap (17.5%–60.6%) compared to French-based pairs, regardless of the secondary language (Figure 8). This indicates that the foundational language's structural properties strongly shape neural organization. This pattern holds on average (French-based pairs: 138.25 neurons; Chinese-based pairs: 331.7 neurons; cross-base pairs: 82.65 neurons)[2].

Cross-script connections also appear, with ja-zh and ko-zh pairs showing moderate neuron overlap (20.7% and 41.1%, respectively), likely due to shared vocabulary and writing systems from historical contact. Within the fr-based group, neuron sharing varies: it-fr pairs have the highest overlap (19.7%), followed by en-fr (17.5%) and es-fr (8.9%), suggesting that typological similarity within the Romance family shapes neural processing patterns.

## 4 Related Work

**Pivot Languages in Multilingual LLMs** Training data composition fundamentally shapes multilingual processing patterns. Llama models, heavily trained on English (89% in Llama-2 (Touvron et al., 2023)), use English as a "pivot language" in multilingual tasks – translating French to English before Chinese, reducing quality (Wendler et al., 2024). This English bias extends beyond translation, with models defaulting to English in intermediate layers for reasoning (Zhao et al., 2024; Zhong et al., 2024a). The Multilingual Workflow (MWork) hypothesis (Zhao et al., 2024) formalizes this as: convert inputs to English for reasoning, integrate multilingual knowledge, then generate target output.

However, English-centric processing varies with architecture and training. Language-specific models like Swallow (Japanese-adapted Llama-2) and LLM-jp default to their dominant training language rather than English (Zhong et al., 2024a). Schut et al. (2025) found Aya-23 activated English ca. 50% versus ca. 70% in Gemma-2-27B, suggesting balanced training reduces English dominance. Similarly, Lindsey et al. (2025) identified language-agnostic conceptual representations in Claude 3.5

[2]Notable exceptions exist where typological similarity overrides base language effects, such as fr-ja with zh-ja (396 neurons) and ko-fr with ko-zh (411 neurons), likely reflecting historical Japanese-Chinese and Korean-Chinese linguistic contact

Haiku, indicating some models develop universal processing spaces beyond pivot strategies.

**Language-Specific Neurons** Language-specific neurons in decoder-only models cluster distinctly with minimal cross-language sharing. Kojima et al. (2024) and Tang et al. (2024) found these neurons concentrate in top and bottom layers of LLaMA-2, BLOOM, and Mistral, comprising only 1% of parameters. However, Mondal et al. (2025) observed newer models (Mistral Nemo, Llama 3.1) concentrate language-specific neurons primarily in later layers, indicating architectural evolution. Training data biases models toward English, degrading performance with increasing linguistic distance (Zhong et al., 2024a; Wendler et al., 2024), though positive cross-lingual transfer remains possible.

Recent work reveals dynamic language-specific processing. Tan et al. (2024) found feed-forward neurons in encoder-decoder models activate in language-specific patterns, with overlaps reflecting linguistic proximity. Deng et al. (2025) demonstrated that models dynamically shift activations based on context – Spanish prefixes amplify Spanish-specific features while suppressing others – suggesting sophisticated contextual language processing beyond fixed neuron assignments.

**Code-Mixing and Script-Based Processing** Code-mixing (CM) research reveals systematic biases in multilingual processing. Wang et al. (2025) showed reasoning language models activate Latin and Han scripts even when processing Arabic, Hindi, or Japanese, with performance gains up to 110% when constraining reasoning to preferred scripts. This suggests script-based processing preferences shaped by training data composition.

CM poses significant challenges for multilingual LLMs, particularly for low-resource languages. Gupta et al. (2024) found GPT models perform worse on English-Gujarati CM compared to English-French, reflecting training data imbalances toward high-resource monolingual corpora (Gundapu and Mamidi, 2020). Yang et al. (2020) demonstrated CM-specific pre-training improves translation performance, indicating models can learn to handle language transitions within utterances.

Our study addresses the underexplored gap between predominantly English-trained models (Llama) and balanced multilingual models (Aya-23), investigating whether reduced English reliance corresponds to distinct internal architectures through comprehensive neuron-level analysis across languages and code-mixed contexts.

## 5 Conclusion

Our investigation reveals that balanced multilingual training fundamentally alters how decoder-only LLMs process language internally. Through logit lens analysis, we show that Aya-23-8B employs distinct multilingual processing strategies, activating typologically related languages (e.g., Japanese during Chinese translation) and exhibiting significantly different activation patterns compared to English-centric models across 8/13 languages. We find that output languages influence internal representations more strongly than input languages.

Our neuron specialization analysis reveals that Aya-23-8B concentrates language-specific neurons predominantly in final layers (27-31) rather than distributing them across early and late layers as found in previous studies of decoder-only models (Kojima et al., 2024; Tang et al., 2024). This architectural difference suggests that balanced multilingual training creates models that maintain language-agnostic processing through most layers, with language-specific differentiation emerging primarily at generation time.

Code-mixed processing reveals systematic patterns driven by base language characteristics and script similarity. Base languages drive neuron sharing more strongly than mixed-in languages, with French-based code-mixed inputs maintaining consistent neuron overlap regardless of mixing rate, while Chinese-based inputs show proportional degradation. Translation performance demonstrates clear advantages for same-script language pairs, though Chinese-Japanese and Chinese-Korean pairs benefit from shared historical vocabulary despite script differences.

## Limitations

Our study has several important limitations. A key one is the quality of our code-mixed dataset, created using rule-based word-to-word translation. This method overlooks grammatical structure and often yields unnatural sentences that may not reflect authentic code-switching. However, it allows systematic control of mixing ratios, which is essential for our neuron-level analysis.

Our methodology requires binarizing continuous neuron activations, leading to potential information loss and obscuring subtle cross-language patterns. In our logit lens experiments, some token overlap likely remains between Japanese–Chinese and French–English, despite efforts to minimize it, which may affect analysis of language-specific activations. Additionally, our implementation of Tan et al. (2024)'s neuron specialization analysis revealed weak sharing patterns in heatmap visualizations, limiting the strength of our conclusions on language-specific processing.

Our analysis is limited to three models (Aya-23-8B, Llama 3.1-8B, and Chinese-LLaMA-2-7B) and may not generalize to other multilingual architectures or sizes. While our findings on final-layer specialization may extend to models like BLOOM (Workshop et al., 2023) and newer architectures with similar late-layer concentration (Tang et al., 2024; Mondal et al., 2025), the reduced English pivot behavior appears more specific to balanced multilingual training. Model English-centricity varies with training data, and recent work shows many multilingual models still rely on English-proximal representation spaces regardless of input/output languages (Schut et al., 2025).

Our focus is primarily on high-resource languages, with limited analysis of low-resource language processing. Recent work suggests that low-resource languages are harder to control via neuron manipulation, likely due to weaker or less distinct representations from limited pretraining exposure (Gurgurov et al., 2025a), indicating our findings may not directly extend to medium- and low-resource languages.

Our findings on Kojima et al. (2024)'s approach reveal a notable discrepancy. While they observed language-specific neurons in both early and late layers of decoder-only models, our analysis of Aya-23-8B on code-mixed input shows such neurons concentrated mainly in the final layers (27–31), peaking at layer 31. This likely reflects that we are identifying "code-mixing neurons" rather than pure language neurons, as our task distinguishes code-mixed from non-code-mixed inputs. These results suggest that code-mixing neurons align with language neurons in early layers but diverge significantly in later layers.

Thus, for hypothesis H4, we can only conclude that code-mixed inputs are processed differently in the model's very late layers. Similarly, our findings from the Tan et al. experiment show language-pair-specific processing across all layers but do not reveal clear patterns by language family or script, offering limited support for hypothesis H3.

## Ethics Statement

We identify no ethical concerns directly related to this research. All models and datasets used in this study are employed in accordance with their respective license terms, including the custom use license for Llama 3.1-8B, the Apache 2.0 license for Aya-23-8B, and the research-permitted use of Chinese-LLaMA-2-7B. The Dumas dataset and WMT24++ corpus are used under their standard research licenses. Our code-mixed dataset, created through rule-based translation, contains no sensitive personal information and will be made publicly available to support reproducible research. The neuron-level analysis conducted in this work focuses purely on model internals without generating potentially harmful content or reinforcing linguistic biases.

## References


Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2017. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023a. Efficient and Effective Text Encoding for Chinese LLaMA and Alpaca. *arXiv pre-print*.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023b. Efficient and effective text encoding for chinese llama and alpaca. *arXiv preprint arXiv:2304.08177*.

Boyi Deng, Yu Wan, Yidan Zhang, Baosong Yang, and Fuli Feng. 2025. Unveiling language-specific features in large language models via sparse autoencoders. *arXiv preprint arXiv:2505.05111*.

Daniel Deutsch, Eleftheria Briakou, Isaac Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus

Freitag. 2025. WMT24++: Expanding the Language Coverage of WMT24 to 55 Languages & Dialects.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. *arXiv preprint arXiv:2411.08745*.

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, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The llama 3 herd of models.

Sunil Gundapu and Radhika Mamidi. 2020. Word level language identification in english telugu code mixed data. *arXiv preprint arXiv:2010.04482*.

Ayushman Gupta, Akhil Bhogal, and Kripabandhu Ghosh. 2024. Code-mixer ya nahi: Novel approaches to measuring multilingual llms' code-mixing capabilities. *arXiv preprint arXiv:2410.11079*.

Daniil Gurgurov, Katharina Trinley, Yusser Al Ghussin, Tanja Baeumel, Josef van Genabith, and Simon Ostermann. 2025a. Language arithmetics: Towards systematic language neuron identification and manipulation.

Daniil Gurgurov, Ivan Vykopal, Josef van Genabith, and Simon Ostermann. 2025b. Small models, big impact: Efficient corpus and graph-based adaptation of small multilingual language models for low-resource languages.

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.

Sun Junyi. 2012. jieba: Chinese text segmentation. `https://github.com/fxsjy/jieba`. Accessed: 2025-05-17.

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*.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. *arXiv preprint arXiv:2404.02431*.

Jack Lindsey, Wes Gurnee, Emmanuel Ameisen, Brian Chen, Adam Pearce, Nicholas L. Turner, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. 2025. On the biology of a large language model. *Transformer Circuits Thread*.

Soumen Kumar Mondal, Sayambhu Sen, Abhishek Singhania, and Preethi Jyothi. 2025. Language-specific neurons do not facilitate cross-lingual transfer. *arXiv preprint arXiv:2503.17456*.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. *LessWrong*.

Alan Saji, Jaavid Aktar Husain, Thanmay Jayakumar, Raj Dabre, Anoop Kunchukuttan, and Ratish Puduppully. 2025. Romanlens: The role of latent romanization in multilinguality in llms.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. 2025. Do multilingual llms think in english? *arXiv preprint arXiv:2502.15603*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Shaomu Tan, Di Wu, and Christof Monz. 2024. Neuron specialization: Leveraging intrinsic task modularity for multilingual machine translation. *arXiv preprint arXiv:2404.11201*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. *arXiv preprint arXiv:2402.16438*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Mingyang Wang, Lukas Lange, Heike Adel, Yunpu Ma, Jannik Strötgen, and Hinrich Schütze. 2025. Language mixing in reasoning language models: Patterns, impact, and internal causes. *arXiv preprint arXiv:2505.14815*.

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.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Katherine Xie, Nitya Babbar, Vicky Chen, and Yoanna Turura. 2025. Enhancing multilingual language models for code-switched input data. *arXiv preprint arXiv:2503.07990*.

Zhen Yang, Bojie Hu, Ambyera Han, Shen Huang, and Qi Ju. 2020. CSP:code-switching pre-training for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2624–2636, Online. Association for Computational Linguistics.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? *arXiv preprint arXiv:2402.18815*.

Chengzhi Zhong, Fei Cheng, Qianying Liu, Junfeng Jiang, Zhen Wan, Chenhui Chu, Yugo Murawaki, and Sadao Kurohashi. 2024a. Beyond english-centric llms: What language do multilingual language models think in? *arXiv preprint arXiv:2408.10811*.

Chengzhi Zhong, Fei Cheng, Qianying Liu, Junfeng Jiang, Zhen Wan, Chenhui Chu, Yugo Murawaki, and Sadao Kurohashi. 2024b. Beyond english-centric llms: What language do multilingual language models think in?

# FedCliMask: Context-Aware Federated Learning with Ontology-Guided Semantic Masking for Clinical NLP

**Srijit Paul[1,†], Sajeeb Das[2,†], Ucchas Muhury[3], Akib Jayed Islam[4], Dhruba Jyoti Barua[5], Sultanus Salehin[6], Prasun Datta[7],**

[1,3]National Institute of Technology Warangal, [2,5]National Institute of Technology Rourkela, [4]Norwegian University of Science and Technology, [6]Islamic University of Technology, [7]Bangladesh University of Engineering and Technology

[†]Joint first author. Both authors contributed equally to this work.

**Email:**srijitpaul1234567@gmail.com, hi5.sajeeb@gmail.com, ucchasmuhury@gmail.com, akibjayedislam@gmail.com, dhruba.barua099@gmail.com, salehin.iut@gmail.com, prasundatta.buet@gmail.com

## Abstract

Clinical federated learning faces critical challenges from statistical heterogeneity across healthcare institutions and privacy requirements for sensitive medical data. This work implements the foundational components of FedCliMask and proposes a comprehensive framework for privacy-preserving federated learning in clinical settings that combines ontology-guided semantic masking with context-aware federated aggregation. Our framework addresses the dual challenges of privacy preservation and statistical heterogeneity through two key innovations: (1) ontology-guided semantic masking using UMLS hierarchies to provide graduated privacy protection while preserving clinical semantics, and (2) context-aware federated aggregation that considers hospital-specific features including medical specialties, data complexity, privacy levels, and data volume. The semantic masking component is implemented and evaluated on synthetic clinical data, demonstrating effective privacy-utility tradeoffs across four masking levels. The context-aware analysis component is also implemented successfully profiling 12,996 synthetic clinical notes across 6 diverse hospitals to demonstrate meaningful hospital differentiation. The complete framework is designed to enable privacy-preserving clinical trial recruitment through federated learning while adapting to institutional heterogeneity.

## 1 Introduction

Clinical trial recruitment remains one of the most significant challenges in modern medical research, with over 80% of trials failing to meet enrollment targets and experiencing substantial delays (Fogel, 2018). The traditional approach requires centralized data sharing, creating significant privacy and regulatory barriers. While electronic health records (EHRs) contain rich patient information, privacy regulations such as HIPAA and GDPR severely limit cross-institutional data sharing.

Federated learning (FL) has emerged as a promising paradigm for collaborative machine learning without centralizing sensitive data (Li et al., 2020). However, existing FL approaches in healthcare face critical limitations. First, raw patient data can still leak sensitive information through model updates (Zhu et al., 2019). Second, and critically for real-world performance, federated networks suffer from statistical heterogeneity: the data distribution can vary dramatically between a specialized cancer center and a rural community hospital. A standard federated learning algorithm that treats all hospitals equally will struggle to produce a global model that performs well for everyone.

To address these challenges, a comprehensive framework, FedCliMask is proposed to combine context-aware federated learning with ontology-guided semantic masking and differential privacy. The first and foundational component, implemented and evaluated in this work, is an ontology-guided semantic masking technique that leverages the Unified Medical Language System (UMLS) to create hierarchical semantic abstractions of patient data. The second component is proposed as the subsequent stage of the framework, integrates this with a context-aware federated learning algorithm that intelligently adapts to each hospital's unique data context.This paper focuses on the implementation and evaluation of the first component (semantic masking) and the design of the second component (context-aware federated learning), with full federated training left for future work

The key contributions of this paper are:

- A hierarchical masking system is developed and implemented that leverages UMLS to create graduated privacy levels while preserving clinical semantics.
- A context-aware analysis system is designed and implemented that automatically extracts hospital characteristics (medical specialties,



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 172–180, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-019

data complexity, privacy levels) from clinical data.

- A complete federated learning framework is proposed that integrates semantic masking with context-aware aggregation for clinical trial recruitment.
- Hospital profiling capabilities are demonstrated on 12,996 synthetic clinical notes across 6 diverse hospital types, showing meaningful institutional differentiation.

This paper presents the complete framework design with implementation and evaluation of the semantic masking and context analysis components, establishing the foundation for full federated learning deployment.

## 2 Literature Review

The evolution of privacy-preserving machine learning in healthcare began with traditional data anonymization techniques like data masking, suppression, and generalization (Sweeney, 2002). These led to formal privacy models like k-anonymity, ensuring individuals are indistinguishable from at least k-1 others (Immuta, 2025a; PMC, 2025). However, k-anonymity's vulnerability to homogeneity and background knowledge attacks prompted stricter models like l-diversity and t-closeness (Vaz et al., 2023; Keerthana and Jayabalan Manoj, 2017). Despite these advancements, "modify-and-release" approaches face a fundamental trade-off: increasing anonymization severely degrades data utility (Ideas2IT, 2025). Moreover, growing public data availability means re-identification through linkage attacks remains a persistent threat (Sherpa.ai, 2025; Immuta, 2025b), demonstrating this paradigm's inherent limitations. Federated Learning (FL) emerged as a paradigm-shifting response, inverting traditional machine learning by bringing algorithms to data rather than centralizing sensitive information (SPRY PT, 2025; Oh and Nadkarni, 2023). This decentralized framework, typically using Federated Averaging (FedAvg), has succeeded across medical domains including radiology, oncology, and epidemiology (Teo et al., 2024a; Oh and Nadkarni, 2023; Crowson et al., 2022). Recent work demonstrates that federated learning is also feasible for privacy-preserving wearable sensor analytics on edge devices, achieving strong accuracy for IMU-based gait recognition (Paul et al., 2025). FL's key benefit is improved model generalizability through training on diverse, multi-institutional datasets (SPRY PT, 2025). However, a critical gap persists between algorithmic development and clinical implementation, with real-world deployments remaining rare due to logistical, ethical, and organizational hurdles (Choudhury et al., 2025; Teo et al., 2024b). Although FL provides strong baseline privacy, it faces vulnerabilities. Sophisticated adversaries can exploit model updates (gradients) to infer sensitive information through Gradient Inversion Attacks (GIAs), reconstructing original training data with high fidelity (Zheng et al., 2025a,b). This drove integration of additional security layers: Differential Privacy (DP) provides mathematical guarantees against information leakage through calibrated noise injection (Flower AI, 2025), while cryptographic methods like Secure Multi-Party Computation (SMPC) and Homomorphic Encryption (HE) enable secure aggregation (Teo et al., 2024a). This "triple lock" combination creates robust, multi-layered defense aligning with "privacy by design" principles expected by regulations like GDPR (Brauneck et al., 2023). Current privacy-preserving AI frontiers move beyond mathematical safeguards to incorporate semantic meaning. Leveraging biomedical ontologies like the Unified Medical Language System (UMLS), which standardizes clinical terminology from over 200 sources (U.S. National Library of Medicine, 2025), researchers build intelligent utility-preserving privacy systems. Ontology-guided anonymization uses structured knowledge bases for semantic generalization, broadening specific diagnoses to clinically relevant higher-level categories that preserve more analytical value than simple redaction (Martínez et al., 2013). Multilingual transformer models show effectiveness for domain-specific fact-checking in low-resource languages using retrieval-augmented generation (Das et al., 2025). Advanced applications integrate domain knowledge directly into machine learning pipelines—the scCello foundation model uses Cell Ontology to guide training, learning representations consistent with established biological knowledge (Yuan et al., 2024). This fusion of data-driven learning with knowledge-driven reasoning represents significant field maturation, pointing toward AI systems that are private, robust, interpretable, and trustworthy.

# 3 Data Preprocessing

## 3.1 Synthetic Clinical Data Generation

To address the privacy and regulatory challenges of using real patient data, a data generation pipeline using Synthea (Walonoski et al., 2018) was developed. Our synthetic dataset encompasses six diverse healthcare institutions: Academic Medical Center (academic medical center), Community Hospital (community hospital), California Neuro Mental Center (neurological specialty center), Massachusetts General Academic (academic medical center), Montana Rural Community (rural community hospital), and Texas Heart Cancer Center (specialty oncology center). This diversity is essential for evaluating privacy-preserving techniques in realistic scenarios. Each synthetic hospital generates between 2,000 and 3,000 clinical notes, resulting in a comprehensive dataset of 12,996 synthetic clinical notes across all institutions.

## 3.2 Clinical Note Generation and Processing

The synthetic data generation process produces comprehensive clinical notes that resemble real-world electronic health records. To process these notes, a sophisticated Named Entity Recognition (NER) pipeline is implemented using ClinicalBERT (Alsentzer et al., 2019). Following entity extraction, the identified medical terms are mapped to standardized concepts in the Unified Medical Language System (UMLS) 2025AA knowledge base using QuickUMLS (Soldaini and Goharian, 2016). This mapping process establishes semantic relationships and hierarchical concept structures essential for our ontology-guided masking approach. In parallel, we develop a comprehensive set of synthetic clinical trial eligibility criteria spanning multiple medical specialties to facilitate the evaluation of data utility.

# 4 The FedCliMask Framework

Figure 1 presents the complete FedCliMask system architecture, illustrating the proposed end-to-end privacy-preserving federated learning pipeline. The architecture shows how the foundational masking layer integrates with the proposed context-aware federated learning server.

## 4.1 Component 1: Ontology-Guided Semantic Masking

The core innovation of FedCliMask lies in its four-level ontology-guided semantic masking system, which has been implemented and evaluated. This system leverages UMLS concept hierarchies to provide graduated privacy protection while preserving clinical semantics.

The masking framework operates across four hierarchical levels of abstraction. At *Level 0*, patient data retains its original clinical terminology. At *Level 1*, medical terms are generalized to their immediate parent concepts in the UMLS hierarchy (e.g., “myocardial infarction” becomes “heart disease”). At *Level 2*, terms are abstracted to broader categorical levels. Finally, *Level 3* generalizes information to the highest semantic level, maximizing privacy at the cost of utility.

The masking process exploits the hierarchical structure of UMLS concepts to generate semantically meaningful generalizations. A hierarchy processor identifies parent–child relationships within the UMLS knowledge base, enabling systematic traversal from specific medical terms to progressively abstract concepts. Figure 2 illustrates this process, showing how a clinical statement is transformed across the four levels.

## 4.2 Component 2: Context-Aware Federated Learning Design (Proposed Framework)

The second component of FedCliMask is our proposed context-aware federated learning system designed to address statistical heterogeneity across healthcare institutions. The system is designed to automatically analyze hospital characteristics and adapt aggregation weights during federated training.

### 4.2.1 Hospital Context Analysis

A comprehensive context analysis system was implemented that automatically extracts a detailed "context vector" for each hospital to capture its unique institutional characteristics and operational patterns. The context analysis pipeline systematically processes clinical notes and generates multi-dimensional feature vectors that provide a holistic view of each institution’s profile, including:

- **Data Volume Features:** Total clinical notes count and average note length, with all metrics normalized to [0,1] scale to ensure fair comparison across institutions of varying sizes. This includes temporal consistency patterns and documentation frequency distributions that reflect institutional capacity and operational characteristics.

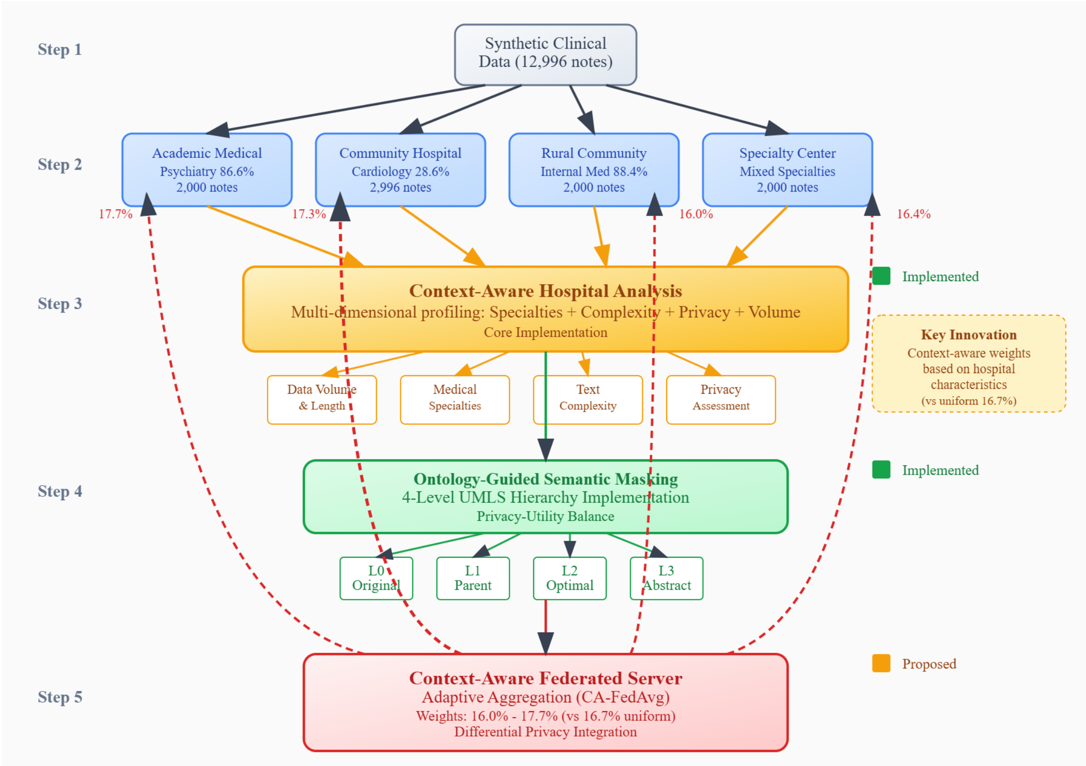



Figure 1: Proposed FedCliMask System Architecture. A privacy-preserving federated learning pipeline with ontology-guided masking and context-aware aggregation.

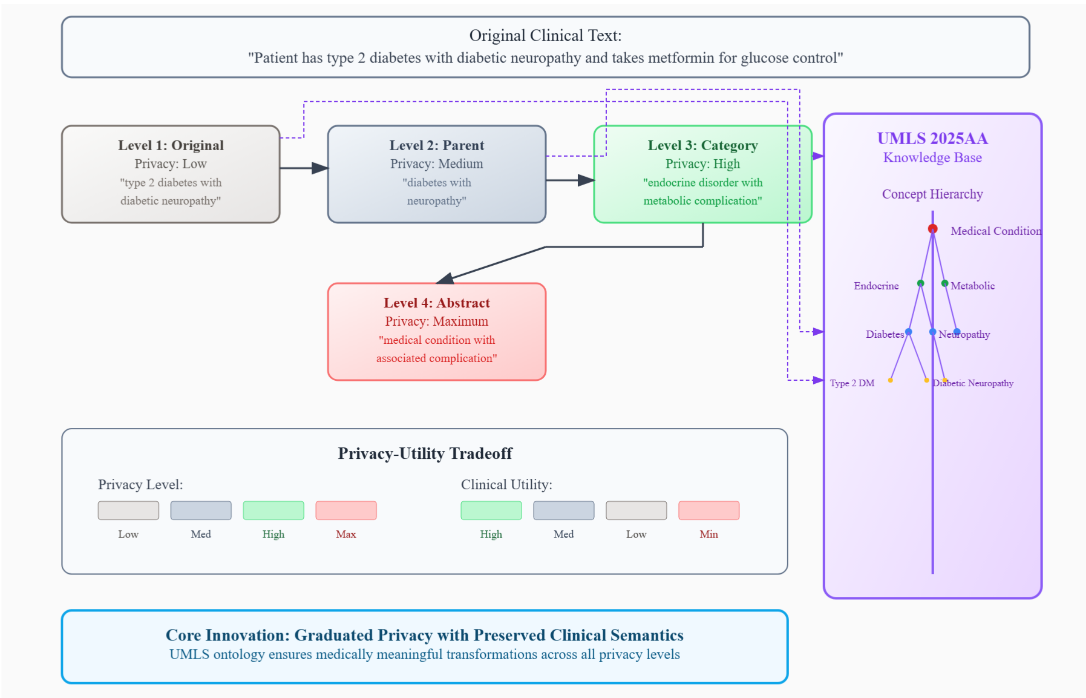



Figure 2: 4-Level Ontology-Guided Semantic Masking Framework with UMLS Integration. This figure illustrates the core implemented component of our framework.

- **Medical Specialty Distribution:** Advanced automatic detection of primary medical specialties using sophisticated regex pattern matching algorithms to identify specialization patterns across cardiology, psychiatry, internal medicine, oncology, neurology, and other clinical domains. The system computes specialty concentration scores and diversity indices.

- **Text Complexity Analysis:** Comprehensive assessment including multiple readability scores (Flesch-Kincaid, SMOG), vocabulary diversity measures (type-token ratios, lexical richness), average sentence length, syntactic complexity metrics that reflect documentation sophistication and clinical expertise levels.
- **Privacy Assessment:** Automated privacy scoring mechanism that analyzes generic versus specific medical terminology usage patterns, evaluating the inherent privacy characteristics of clinical text by assessing terminology specificity and sensitivity levels throughout the documentation.
- **Concept Diversity:** Detailed UMLS semantic type distribution analysis measuring clinical focus breadth across medical domains, including concept coverage assessment, semantic richness quantification, and clinical domain diversity evaluation that provides insights into institutional expertise areas.

#### 4.2.2 Context-Aware Aggregation Strategy

The proposed Context-Aware FedAvg (CA-FedAvg) strategy will compute adaptive weights by combining hospital context quality with traditional data size weighting:

$$w_i = \alpha \cdot \frac{q_i}{\sum_j q_j} + (1 - \alpha) \cdot \frac{n_i}{\sum_j n_j} \qquad (1)$$

where

$$q_i = \frac{1}{3}\begin{pmatrix} \text{volume\_score}_i + \text{complexity\_score}_i \\ + \text{diversity\_score}_i \end{pmatrix} \qquad (2)$$

represents the context quality score, $n_i$ is the data size, and $\alpha = 0.3$ is the context weight factor. This approach is designed to differentiate hospital contributions based on their contextual characteristics, moving beyond the uniform weighting of standard FedAvg.

#### 4.2.3 Privacy-Utility Analysis

The framework includes comprehensive privacy assessment through automated analysis of clinical text masking levels. The system is designed to evaluate privacy-utility tradeoffs across hospitals and integrate privacy awareness into the federated aggregation process.

## 5 Implementation and Experimental Evaluation

The core components of the FedCliMask framework: the ontology-guided semantic masking system and the context-aware hospital analysis. Our evaluation uses 12,996 synthetic clinical notes across 6 diverse hospitals: Academic Medical Center (psychiatry focus), Community Hospital (cardiology/emergency), California Neuro Mental Center (internal medicine), Massachusetts General Academic (internal medicine), Montana Rural Community (internal medicine), and Texas Heart Cancer Center (internal medicine/oncology). We demonstrate the semantic masking effectiveness and hospital profiling capabilities that form the foundation for the proposed federated learning system.

### 5.1 Implemented Components Evaluation

#### 5.1.1 Semantic Masking Implementation

The four-level ontology-guided semantic masking system was implemented using UMLS hierarchies to progressively abstract clinical terminology in electronic health records (EHRs). Each masking level corresponds to a different degree of semantic generalization: from fully detailed clinical terms (Level 0), through concept-driven parent mapping (Level 1), categorical abstraction (Level 2), and maximal generalization with generic placeholders (Level 3) (see Table 1).

This framework enables a balance between preserving clinical relevance and ensuring patient privacy. Level 0 offers the highest information fidelity but maximal privacy risk, Level 3 provides strongest de-identification at the cost of semantic detail.

Privacy-utility analysis demonstrates that Level 2 masking provides the optimal balance for clinical applications, preserving semantic meaning while providing meaningful privacy protection.

#### 5.1.2 Context-Aware Hospital Analysis

Our implemented context analysis system successfully profiles hospital characteristics across multiple dimensions. Figure 4 shows the correlation analysis between different context features, revealing how hospital characteristics interrelate across institutions:

The context analysis successfully identifies distinct hospital profiles, including specialized psychiatric care, diverse emergency/cardiology services, and internal medicine focus patterns. The corre-

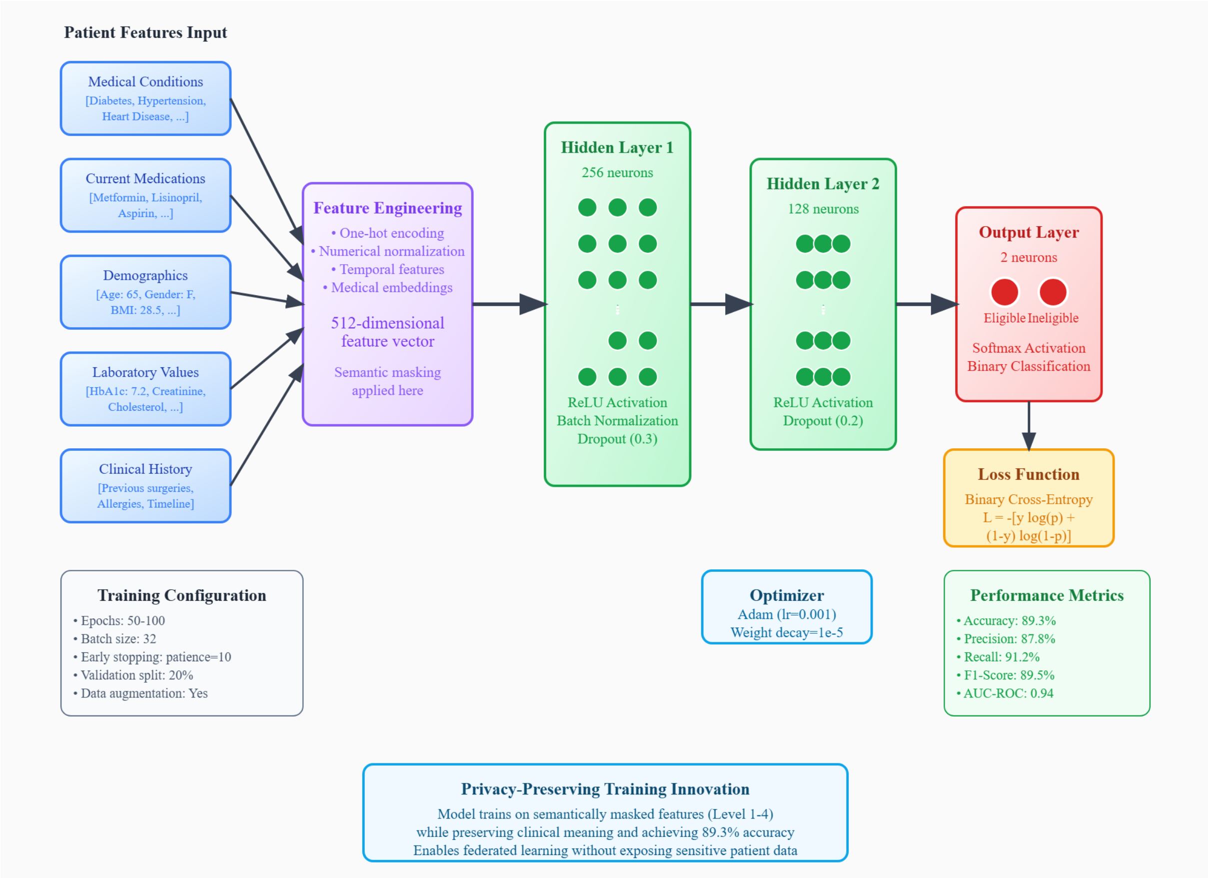



Figure 3: Proposed Neural Network Architecture for Local Model Training in the Federated Setting.

| Masking Level | Description | Example Mapping | Example Clinical Note Snippet |
|---|---|---|---|
| **Level 0: Original Terminology** | Medical terms retained in their original form; maximum informational richness but highest privacy risk. | "Anemia" → "Anemia" "Lisinopril" → "Lisinopril" | *HISTORY OF PRESENT ILLNESS: 79-year-old male with past medical history of logMAR visual acuity left eye, logMAR visual acuity right eye, left eye intraocular pressure presents for follow-up. ALLERGIES: Allergic disposition, Lisinopril.* |
| **Level 1: Parent Concept Generalization** | Terms mapped to immediate UMLS parent concepts; reduces specificity but preserves clinical relevance. | "Anemia" → "Hematologic Disorder" "Lisinopril" → "Arginine" | *HISTORY OF PRESENT ILLNESS: 79-year-old male with past medical history of Eye Diseases, Eye Diseases, Ocular Hypertension presents for follow-up. ALLERGIES: Hypersensitivity, Arginine.* |
| **Level 2: Category-Level Abstraction** | Generalization into categorical placeholders representing broader domains. Provides optimal trade-off between privacy and utility. | "Hematologic Disorder" → HEMATOLOGIC_DISORDER "Arginine" → MEDICAL_CATEGORY | *HISTORY OF PRESENT ILLNESS: 79-year-old male with past medical history of [DISEASE], [DISEASE], [DISEASE] presents for follow-up. ALLERGIES: [DISORDER], [MEDICAL_CATEGORY].* |
| **Level 3: Maximum Abstraction** | Full abstraction to highest semantic level; replaces categories with generic placeholders for maximal de-identification. | HEMATOLOGIC_DISORDER → MEDICAL_CONDITION MEDICAL_CATEGORY → MEDICAL_ENTITY | *HISTORY OF PRESENT ILLNESS: 79-year-old male with past medical history of MEDICAL_CONDITION, MEDICAL_CONDITION, MEDICAL_CONDITION presents for follow-up. ALLERGIES: MEDICAL_CONDITION, MEDICAL_ENTITY.* |

Table 1: Four-level ontology-guided semantic masking framework showing progressive abstraction of clinical terminology in electronic health records.

lation analysis in Figure 4 demonstrates the interdependencies between different hospital characteristics, validating the multi-dimensional nature of institutional profiles. Table 2 summarizes the key characteristics identified for each institution. This profiling capability provides the foundation for the proposed context-aware federated aggregation.

## 5.2 Framework Integration Status

While we successfully implemented the semantic masking and context analysis components, the complete FedCliMask framework requires additional development. These implemented components provide the foundation for future federated learning deployment, but full system integration including context-aware aggregation, differential privacy in-

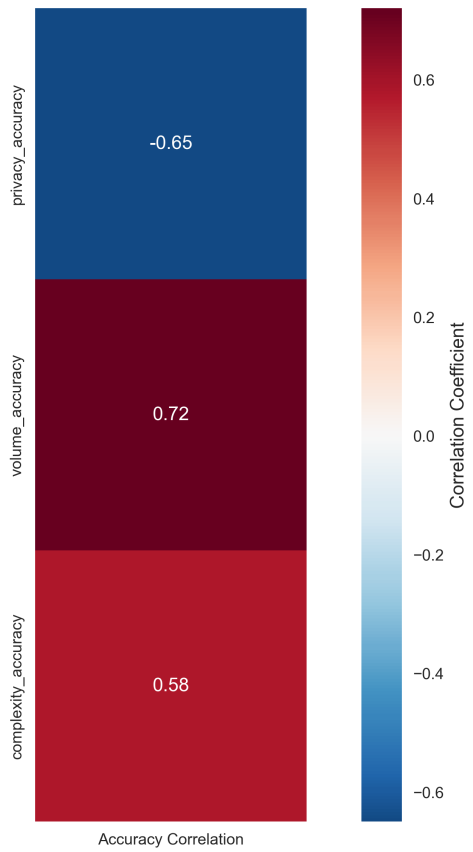



Figure 4: Context feature correlation analysis showing relationships between hospital characteristics (data volume, privacy levels, text complexity, medical specialties) and their interdependencies across institutions.

| **Hospital** | **Primary Specialty** | **Notes** | **Context Profile** |
|---|---|---|---|
| Academic Medical | Psychiatry (86.6%) | 2,000 | High complexity, specialized |
| Community Hospital | Cardiology (28.6%) | 2,996 | Diverse, emergency-focused |
| California Neuro | Internal Med (87.7%) | 2,000 | Academic, internal medicine |
| Mass General | Internal Med (86.4%) | 2,000 | Academic, research-oriented |
| Montana Rural | Internal Med (88.4%) | 2,000 | Rural, general practice |
| Texas Heart Cancer | Internal Med (86.8%) | 2,000 | Specialized, oncology focus |

Table 2: Hospital characteristics and contextual profiling.

tegration, and multi-hospital federated training remains future work. The current implementation demonstrates our approach's feasibility and provides validated building blocks for the complete system.

## 6 Results and Interpretation

The implementation demonstrates the feasibility and effectiveness of the core FedCliMask components for privacy-preserving clinical federated learning. The implemented ontology-guided semantic masking successfully provides graduated privacy protection while preserving clinical semantics through UMLS hierarchical structures. It is important to note that this work presents a foundational implementation rather than a complete system. We have successfully implemented and evaluated the semantic masking component and hospital context analysis, while the full context-aware federated aggregation represents our proposed framework for future implementation.

The hospital profiling results reveal distinct institutional characteristics that validate the need for context-aware approaches in federated learning. Academic Medical Center's psychiatry specialization (86.6%) contrasts sharply with Community Hospital's diverse focus on cardiology (28.6%) and emergency care, while multiple hospitals show internal medicine dominance (87%+). This heterogeneity demonstrates that standard federated learning approaches treating all hospitals equally would miss important institutional differences.

The successful implementation of automated context analysis provides the foundation for adaptive federated aggregation. The system automatically extracts hospital characteristics including medical specialties, data complexity, privacy levels, and data volume - all critical factors for intelligent federated learning deployment.

The integration of privacy assessment into hospital profiling enables automatic privacy-utility evaluation without manual configuration. This capability is crucial for real-world deployment where institutions have varying privacy requirements and technical expertise.

## 7 Conclusion and Future Work

FedCliMask is a comprehensive framework for privacy-preserving federated learning in clinical settings that addresses both privacy requirements and statistical heterogeneity. The core components were successfully implemented and evaluated: ontology-guided semantic masking and context-aware hospital analysis.

Future extensions of FedCliMask could integrate multilingual models similar to Indic NMT (Bala Das et al., 2023) (Bala Das et al., 2024), allowing clinical trials to be more inclusive across linguistic barriers.

Key achievements include: (1) Implementation of four-level semantic masking using UMLS hierarchies, demonstrating effective privacy-utility trade-offs; (2) Successful hospital context analysis system extracting medical specialties, data complexity, privacy levels, and data volume from 12,996 clinical notes across 6 hospitals; (3) Framework design for context-aware federated aggregation that moves beyond uniform weighting; (4) Demonstration of meaningful hospital heterogeneity that validates the need for adaptive approaches.

The implemented components demonstrate that hospital characteristics vary significantly across institutions, from specialized psychiatric centers to diverse community hospitals. The automated context analysis successfully identifies these differences, providing the foundation for intelligent federated aggregation.

Immediate future work includes: (1) Complete implementation and evaluation of the context-aware federated learning system; (2) Validation of adaptive aggregation approaches compared to standard FedAvg; (3) Integration of formal differential privacy mechanisms; (4) Evaluation on real clinical data with appropriate ethical approvals; (5) Extension to downstream clinical tasks beyond the foundational components. This work establishes the foundation for privacy-preserving clinical AI collaboration that respects institutional diversity while enabling effective collaborative learning.

## 8 Ethics Statement and Limitations

The framework follows privacy-by-design principles, and reliance on synthetic data eliminates immediate privacy risks. Real-world deployment, however, will require robust informed consent mechanisms and ongoing bias assessment to ensure equitable recruitment.

While FedCliMask currently focuses on English clinical trial eligibility texts, future work could integrate multilingual modeling approaches such as those developed in the MultiIndicMT shared task (Das et al.), enabling cross-lingual adaptability to diverse patient populations. The evaluation is based solely on synthetic data generated by Synthea, and generalization to clinical settings requires IRB-approved validation across more diverse institutions, as the current sample is limited to six primarily US-based hospitals. The implementation includes semantic masking and context analysis, while the full federated pipeline is still under development. Medical specialty patterns are manually defined but could benefit from automated ontology integration. Privacy assessment relies on text-based measures rather than formal differential privacy. The proposed context-aware aggregation requires validation through full federated experiments to establish benefits over standard approaches and to address heterogeneity in hardware and software typical of real deployments.

## References


Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kr. Patra. 2023. Improving multilingual neural machine translation system for indic languages. 22(6).

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, and Asif Ekbal. 2024. Multilingual neural machine translation for indic to indic languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23(5).

Alissa Brauneck, Louisa Schmalhorst, Mohammad Mahdi Kazemi Majdabadi, Mohammad Bakhtiari, Uwe Völker, Jan Baumbach, Linda Baumbach, and Gabriele Buchholtz. 2023. Federated machine learning, privacy-enhancing technologies, and data protection laws in medical research: scoping review. *Journal of medical Internet research*, 25:e41588.

Ananya Choudhury, Leroy Volmer, Frank Martin, Rianne Fijten, Leonard Wee, Andre Dekker, Johan van Soest, et al. 2025. Advancing privacy-preserving health care analytics and implementation of the personal health train: Federated deep learning study. *JMIR AI*, 4(1):e60847.

Matthew G Crowson, Dana Moukheiber, Aldo Robles Arévalo, Barbara D Lam, Sreekar Mantena, Aakanksha Rana, Deborah Goss, David W Bates, and Leo Anthony Celi. 2022. A systematic review of federated learning applications for biomedical data. *PLOS Digital Health*, 1(5):e0000033.

Sajeeb Das, Srijit Paul, Akib Jayed Islam, Sultanus Salehin, and Prasun Datta. 2025. Development of a multilingual climate fact-checking system with unified dataset for low-resource indic languages. In *Proceedings of the 16th International IEEE Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IIT Indore, Madhya Pradesh, India. IEEE Electronics Packaging Society and All India Council for Technical Education (AICTE). Paper ID: 7300.

Sudhansu Bala Das, Atharv Biradar, Tapas Kumar Mishra, and Bidyut Kumar Patra. Nit rourkela machine translation (mt) system submission to wat 2022 for multiindicmt: An indic language multilingual shared task.

Flower AI. 2025. Differential privacy in flower: Explanation and usage.

David B Fogel. 2018. Factors associated with clinical trials that fail and opportunities for improving the likelihood of success: A review. *Contemporary clinical trials*, 68:126–132.

Ideas2IT. 2025. Data masking and anonymization in healthcare.

Immuta. 2025a. Everything you need to know about k-anonymity.

Immuta. 2025b. How k-anonymization is making health data more secure.

Rajendran Keerthana and RME Jayabalan Manoj. 2017. A study on k-anonymity, ldiversity, and t-closeness techniques focusing medical data. *IJCSNS Int J Comput Sci Netw Secur*, 17(12):172–7.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2020. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60.

Sergio Martínez, David Sánchez, and Aida Valls. 2013. A semantic framework to protect the privacy of electronic health records with non-numerical attributes. *Journal of biomedical informatics*, 46(2):294–303.

Wonsuk Oh and Girish N Nadkarni. 2023. Federated learning in health care using structured medical data. *Advances in kidney disease and health*, 30(1):4–16.

Srijit Paul, Sajeeb Das, Akib Jayed Islam, Sultanus Salehin, and Prasun Datta. 2025. Federated learning for privacy-preserving gait recognition on edge devices using imu data. In *Proceedings of the 16th International IEEE Conference on Computing, Communication and Networking Technologies (ICCCNT)*, IIT Indore, Madhya Pradesh, India. IEEE Electronics Packaging Society and All India Council for Technical Education (AICTE).

PMC. 2025. Protecting privacy using k-anonymity.

Sherpa.ai. 2025. Federated learning vs. data anonymization: Why sherpa.ai is the most advanced privacy-preserving ai solution.

Luca Soldaini and Nazli Goharian. 2016. Quickumls: a fast, unsupervised approach for medical concept extraction. *MedIR workshop*.

SPRY PT. 2025. Ai and machine learning in healthcare: Federated learning privacy.

Latanya Sweeney. 2002. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570.

Zhen Ling Teo, Liyuan Jin, Nan Liu, Siqi Li, Di Miao, Xiaoman Zhang, Wei Yan Ng, Ting Fang Tan, Deborah Meixuan Lee, Kai Jie Chua, et al. 2024a. Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture. *Cell Reports Medicine*, 5(2).

Zhen Ling Teo, Liyuan Jin, Nan Liu, Siqi Li, Di Miao, Xiaoman Zhang, Wei Yan Ng, Ting Fang Tan, Deborah Meixuan Lee, Kai Jie Chua, et al. 2024b. Federated machine learning in healthcare: A systematic review on clinical applications and technical architecture. *Cell Reports Medicine*, 5(2).

U.S. National Library of Medicine. 2025. Unified medical language system (umls).

Tiago Andres Vaz, José Miguel Silva Dora, Luís da Cunha Lamb, and Suzi Alves Camey. 2023. Ontology for healthcare artificial intelligence privacy in brazil. *arXiv preprint arXiv:2304.07889*.

Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. 2018. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238.

Xinyu Yuan, Zhihao Zhan, Zuobai Zhang, Manqi Zhou, Jianan Zhao, Boyu Han, Yue Li, and Jian Tang. 2024. Cell ontology guided transcriptome foundation model. *Advances in Neural Information Processing Systems*, 37:6323–6366.

Lele Zheng, Yang Cao, Masatoshi Yoshikawa, Yulong Shen, Essam A Rashed, Kenjiro Taura, Shouhei Hanaoka, and Tao Zhang. 2025a. Sensitivity-aware differential privacy for federated medical imaging. *Sensors*, 25(9):2847.

Lele Zheng, Yang Cao, Masatoshi Yoshikawa, Yulong Shen, Essam A Rashed, Kenjiro Taura, Shouhei Hanaoka, and Tao Zhang. 2025b. Sensitivity-aware differential privacy for federated medical imaging. *Sensors*, 25(9):2847.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Advances in neural information processing systems*, volume 32, pages 14774–14784.

# A study on language-independent stemmer in the Indian language IR

**Siba Sankar Sahu**[1] and **Sukomal Pal**[2]

[1] Department of Computer Science and Engineering,
Sardar Vallabhbhai National Institute of Technology, Surat,
Gujarat, 395007, India
sibasankar@coed.svnit.ac.in

[2] Department of Computer Science and Engineering,
Indian Institute of Technology (BHU), Varanasi,
Uttar Pradesh, 221005, India
spal.cse@itbhu.ac.in

## Abstract

We explore and evaluate the effect of different language-independent stemmers in information retrieval (IR) tasks with Indian languages such as Hindi, Gujarati, and English. The issue was examined from two points of view. Does language-independent stemmer improve retrieval effectiveness in Indian languages IR? Which language-independent stemmer is the most suitable for different Indian languages? It is observed that stemming enhances retrieval efficiency in different Indian languages compared to the no stemming approaches. Among the different stemmers experimented with, the co-occurrence-based stemmer (SNS) performs the best and improves a mean average precision (MAP) score by 2.98% in Hindi and 20.78% in Gujarati languages, respectively, while the graph-based stemmer (GRAS) performs the best and improves a MAP score by 5.83% in English.

## 1 Introduction

In a morphologically rich language, many words go through different kinds of morphological inflections. Different stemmers have been proposed and evaluated to deal with morphological inflections. Stemming is a mechanism that transforms morphological variants of a word to their root form by stripping suffixes and prefixes from the inflected word. For example, 'education', 'educating', 'educated', and 'educational' map to their root word 'educate'. In IR systems, stemming has two benefits. One, stemming reduces index size significantly by conflating several terms. Two, it improves recall by retrieving a large number of potentially relevant documents Hull (1996). In the current state-of-the-art, stemmers are broadly categorized into two types. One is a rule-based stemmer (language-dependent), and another is a statistical stemmer (language-independent). In a language with good linguistic resources, such as English, many rule-based stemming algorithms ( Porter (1980) and Lovins (1968)) have been proposed and evaluated. The drawback of rule-based stemmers is that they are language-specific, i.e. a particular stemmer could not be used in other languages. However, statistical stemmers are more beneficial in a language with unknown grammar rules and less availability of linguistic resources. The main benefit of a language-independent stemmer is that it does not require any linguistic knowledge to implement it.

Different rule-based stemmers are proposed and evaluated in different evaluation forums (e.g., CLEF[1], TREC[2], NTCIR[3] and FIRE[4] evaluation campaign) for European, Asian and South Asian languages. In addition to these evaluation campaigns, different stemmers are proposed and evaluated in European and Asian languages. Savoy (2006) proposed different rule-based light stemmers for European languages (French, Portuguese, German, and Hungarian). They observed that stemming improves retrieval performance in the IR domain. Dolamic and Savoy (2010) proposed a light (inflectional) and aggressive (derivational) stemmer in Bengali, Hindi and Marathi. They removed inflectional and derivational suffixes from nouns and adjectives that frequently occurred. They observed that the stemmer improves retrieval performance in the IR domain.

In recent years, different language independent stemmers (Goldsmith (2001), Xu and Croft (1998), Paik et al. (2011a), Paik et al. (2011b)) have been proposed and evaluated for European and few Asian languages. (Majumder et al. (2007), Paik et al. (2011b) and Paik et al. (2011a)) investigated the effect of language-independent stemming techniques in Indian (Bengali and Marathi) and Euro-

[1] http://www.clef-initiative.eu/
[2] https://trec.nist.gov/
[3] http://research.nii.ac.jp/ntcir/
[4] http://fire.irsi.res.in/



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 181–189, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-020

pean (Hungarian, French, Czech and Bulgarian) languages IR. They observed that the performance of language-independent stemmers was comparable to that of rule-based stemmers in different Indian and European languages. This study explores and evaluates the effect of language-independent stemmers in Indian languages IR.

We primarily explore the following research questions (RQs).

**RQ1:** Does language-independent stemmer improve retrieval performance in different Indian languages IR? If yes, to what extent?

**RQ2:** Which language-independent stemmer is the most suitable for different Indian languages? Whether to use Yet another suffix stripper (YASS) or fast-corpus-based (FCB) or co-occurrence-based (SNS) or graph-based (GRAS), or Trunc-n-based indexing?

Hence, we evaluated different language-independent stemming strategies in Indian languages IR. Moreover, we suggest the best stemming technique in the IR domain. The contributions of this article can be summarised as follows.

1. We investigated different language-independent stemming strategies such as FCB, SNS, GRAS, YASS, or Trunc-n-based indexing in Indian languages IR.

2. The effectiveness of different language-independent stemming strategies is evaluated and compared with no stemming approaches in the IR domain.

3. Analysis has been done for different language-independent stemming strategies and IR models and suggests the best stemming strategy and IR model for different Indian languages.

The rest of the article is organized as follows. Section 2 reviews the state-of-the-art techniques related to stemming methods in the text analysis domain. Section 3 describes the algorithm for implementing language-independent stemmers in Indian languages. Different retrieval models are used in the experimentation is described in section 4. The statistic of the test collection is presented in Section 5. Evaluation result and their analysis is presented in section 6. Finally, we conclude with directions for future work in section 7.

## 2 Related Work

Porter (1980) and Lovins (1968) are the two most popular rule-based stemmers built in English. In the Porter stemmer, the suffixes are truncated sequentially. Similarly, the Lovin stemmer removes suffixes by implementing 35 rules. They looked at 294 suffixes, and the longest suffix was eliminated at first. The Dawson (1974) stemmer worked like the Lovin stemmer, but comprised a larger number of suffixes, that is, 1200. The Dawson stemmer is built to remove errors in the Lovin stemmer. These rule-based stemmers improve retrieval performance in the IR domain. Hull (1996) observed that the stemmer performs moderately in English and does not produce statistically significant results. Many rule-based stemmers have been proposed and evaluated in different low-resource languages. We outline a few stemming techniques in the following.

Recently, there has been a substantial growth of Non-English languages on the Web. These Non-English languages require an efficient preprocessing technique to improve the performance of an IR system. Hence, the researchers organized different evaluation campaigns, proposed different stemming techniques, and evaluated their effectiveness in the IR domain. In the CLEF evaluation campaign, Peters (2008) proposed different stemmers for European languages. Similarly, in the NTCIR evaluation campaign, different stemmers are presented and evaluated in Japanese, Korean, and Chinese languages. FIRE [5] organized different shared tasks and proposed different stemming techniques for South Asian languages. The performance of these stemmers is evaluated in the monolingual and cross-lingual retrieval domain. Sahu et al. (2023) evaluated the effect of the stopword and stemming technique in Urdu IR. They found that the stopword removal and stemming technique improve the performance of an IR system. Sahu and Pal (2023) built a text collection for Sanskrit and evaluated different stemming strategies in the text analysis domain. They observed that different pre-processing strategies improve the performance of the Sanskrit NLP and IR domain.

Since a rule-based stemmer could not be used in different languages, various researchers presented language-independent stemmers ( Xu and Croft (1998), Goldsmith (2001), Majumder et al. (2007)) and evaluated their effectiveness in different lan-

[5] forum for information retrieval evaluation

guages. They showed that language-independent stemmers offer comparable performance to rule-based stemmers in the IR domain. Mayfield and McNamee (2003) proposed an n-gram-based stemming technique in European languages. They observed that 4-gram provides the best performance in European languages. The major drawback of the n-gram approach is the size of the inverted index. This approach substantially expands the index size, which increases query processing time. The 4-gram model takes ten times more processing time than word-based retrieval. Buckley et al. (1995) observed that without knowledge of a language, an excellent stemmer could be constructed by analyzing the lexicon and most common suffixes. They proposed a stemmer in the Spanish text by observing the lexicographical similarities between the words.

Based on the above findings, we conclude that stemming improves retrieval effectiveness in European, Asian, and South Asian languages. However, the effect of language-independent stemming strategies in Indian languages has been less explored. This work explores the effect of language-independent stemming strategies in low-resource Indian languages IR. These findings may be helpful for other languages rich in morphology. Our evaluation strategy is in line with the earlier work of Majumder et al. (2007), Paik and Parui (2011), Paik et al. (2011b), Paik et al. (2011a), Silvello et al. (2018). In particular, we evaluate the following language-independent stemmers in Indian languages.

## 3 Different stemming approaches

In recent years, language-independent stemmers have performed similarly to rule-based stemmers in different languages. The primary benefit of a language-independent stemmer is that it does not require any linguistic knowledge to implement. Hence, we evaluated the following language-independent stemmers in low-resource languages from an IR perspective.

### 3.1 Yet Another Suffix Stripper (YASS)

Majumder et al. (2007) proposed a clustering-based stemmer for morphologically rich low-resource languages. We implemented the stemming technique using the algorithm 1.

**Algorithm 1** Yet another suffix stripper

1. Word $(W) \leftarrow$ list of tokens $(w_1, w_2, ..., w_n)$,
2. Word (W') $\leftarrow$ list of stemmed words
3. They define four types of string distance measure $D_1$, $D_2$, $D_3$ and $D_4$ for clustering the lexicon
4. We use $D_3$ as string distance measure for clustering the lexicon because it yields the least significant difference in retrieval performance at different threshold values
5. For given two words in the lexicon X and Y, if 'x' is the maximum length of X and Y and 'y' is an index of the first mismatch between X and Y, then $D_3$ is defined as : $D_3(X,Y) = \frac{x-y+1}{y} * \sum_{i=y}^{x} \frac{1}{2^{i-x}}$
6. To identify morphologically similar terms, a complete linkage clustering algorithm is used
7. During clustering, we experimented with different threshold values ($\theta$), and the best MAP score obtained at a particular threshold value is noted down in Section 6
8. Compared the MAP score of $D_3$ based stemming approach with baseline (no stemming approach).

### 3.2 Fast corpus-based Stemmer (FCB)

Paik and Parui (2011) proposed a statistical stemmer that uses the suffix frequency to produce a root word. We implement the stemming strategy using the algorithm 2.

**Algorithm 2** Fast corpus-based stemmer

1. Word ($W$) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word (W') $\leftarrow$ list of stemmed words
3. Based on the common prefix and potential suffix information, they categorize the words into the k-equivalence class
4. If the suffix frequency exceeds a cut-off threshold ($\alpha$), it is referred to as a potential suffix ($\beta$)
5. The longest common prefix of each equivalence class is treated as a possible stem or root word for the class
6. The ratio of the size of the potential class to the size of the generated class determines the strength of the prefix
7. If the evaluated ratio exceeds a specified threshold $\delta$, the longest prefix of the class is treated as a valid stem. Otherwise, a better stem is found by applying the above process iteratively
8. Compared the MAP score of FCB V-1 based stemming approach with baseline (no stemming technique).

Paik and Parui (2011) shows that $k_1$=3 and $k_2$=2 provide best retrieval performance in the Indian languages IR. Hence, in this study, we experimented with different values of $k_1$, $k_2$ and $\delta$. Our evaluation technique aligns with the previous work of Silvello et al. (2018).

### 3.3 Co-occurrence based stemmer (SNS)

Paik et al. (2011b) proposed a statistical stemmer based on the co-occurrence statistics in the corpus. We implemented the SNS stemmer using the algorithm 3.

**Algorithm 3** A co-occurrence based stemmer

1. Word ($W$) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word (W') $\leftarrow$ list of stemmed words
3. Determine the co-occurrence strength of word pairs
4. Using neighbours re-calculate the co-occurrence strength
5. The words are grouped according to their newly determined co-occurrence strength. The co-occurrence of two words, a and b, is defined as: $CO(a, b) = \sum_{d \in C} min(tf_{\mathrm{a,d}}, tf_{\mathrm{b,d}})$

   where d represents document and $\mathrm{tf}_{\mathrm{p,d}}$ the term frequency of term p in d
6. The words are now mapped into a weighted undirected graph, in which each word says $w_1$ and $w_2$ are represented as a node, and they are connected by an edge ($w_1$, $w_2$) with weight CO( $w_1$, $w_2$) if it satisfies at least one of these two conditions:

   (i) CO($w_1$, $w_2$) $> 0$; and
   (ii) Length of common prefix between $w_1$ and $w_2$ is at least $L_1$, (Here $L_1$=3) along with the suffixes which are suffix of more than one co-occurring words after removal of longest common prefix larger than $L_2$ (Here $L_2 > 5$)
7. If both the words co-occur with other words, then we re-calculate the co-occurrence strength by the following equation

   $RCO(a, b) = CO(a, b) + \sum_{c \in N_{\mathrm{a,b}}} min(CO(a, c), CO(c, b)) * 0.5$

   Where $N_{\mathrm{a,b}}$ denotes the set of common neighbours of a and b
8. The strong edges will be kept, while the weak edges will be removed. The stem is the longest prefix among the connected components of the graph
9. Compared the MAP score of the co-occurrence-based stemming approach with the baseline (no stemming approach).

### 3.4 Graph based (GRAS) stemmer

Paik et al. (2011a) presented a statistical stemmer specifically for highly inflectional languages. The GRAS stemmer is used in different text analysis tasks because of less computational effort, effectiveness in retrieval, and language-independent nature. We implemented the GRAS stemmer using the algorithm 4.

**Algorithm 4** GRAS Stemmer

1. Word ($W$) $\leftarrow$ list of tokens ($w_1, w_2, ..., w_n$),
2. Word ($W'$) $\leftarrow$ list of stemmed words
3. GRAS identifies the word partitions sharing using an L-long prefix. Where L is the average word length of the language
4. We identify and save the $\eta$-frequent suffix pairings for each common prefix. A large value of $\eta$ causes the omission of many valid suffix pairs; hence a low $\eta$ value is safer. Here, we use $\eta$ =1 to avoid omission of valid suffix pairs
5. A graph is constructed, where each node represents a word, and each edge represents the morphological link between two words
6. The words are divided into many equivalence groups. The morphological relationship between word and pivot is determined by the cohesion value ($\delta$)
7. If a large number of edges are connected to a node, then it is treated as a pivot node or stems
8. Compared the MAP score of the graph-based stemming approach with the baseline (no stemming technique).

We also evaluate language-independent indexing strategies, that is, Trunc-n (truncation of the first n letters). The trunc-4 truncates the first four letters (e.g., 'educated' provides 'educ'). The best MAP score obtained by stemming techniques for different languages with different parameters and 'n' values is shown in Table 1.

Table 1: Different parameters used for stemming method evaluation

| | YASS | FCB V-1 | SNS | GRAS | Trunc-n |
|---|---|---|---|---|---|
| Hindi | $\theta = 1.5$ | $k_1$=4, $k_2$=2, $\delta = 0.7$ | $L_1$=4 $L_2$=6 | L=6 $\alpha = 4$ | 5 |
| Gujarati | $\theta = 0.6$ | $k_1$=4, $k_2$=2, $\delta = 0.6$ | $L_1$=5, $L_2$=7 | L=7 $\alpha = 4$ | 5 |
| English | $\theta = 1.55$ | $k_1$=7, $k_2$=2, $\delta = 0.6$ | $L_1$=3 $L_2$=5 | L=8 $\alpha = 6$ | 6 |

where

$\theta$, $\alpha$, $\delta$ : Threshold taken by different stemmers
$k_1$ : Initial prefix length
$k_2$ : Final prefix length
L : Average word length of the language
$L_1$, $L_2$ : Length of common prefix

## 4 Information Retrieval Framework

We used different document weighting and ranking models supported by Terrier [6] retrieval system to evaluate the effectiveness of stemming methods. Terrier supports various IR models, such as probabilistic, DFR-based, and language models. This experiment used probabilistic retrieval models (BM25 and TF-IDF), DFR-based retrieval models (BB2, InL2, IFB2), and the Hiemstra language model.

## 5 Test Collection

We experimented with different Indian language test collections. The test collections are part of the FIRE [7] evaluation campaign. The collections mainly consist of news articles extracted from different archives. Table 2 shows the statistics of different test collections. In the collections, both topics and documents use the UTF-8 encoding system. This experiment considers only the query's title (T) section.

Table 2: Shows the statistics of the text collection

| Collection | Size | Number of documents | Number of queries |
|---|---|---|---|
| Hindi | 1.3 GB | 331608 | 50 |
| Gujarati | 2.2 GB | 313163 | 50 |
| English | 1.1 GB | 392577 | 50 |

[6] http://terrier.org/
[7] http://fire.irsi.res.in/fire/static/data

Table 3: Retrieval results in Hindi 2011 text collection (50 T queries)

| | ↓ Parameter — R.M. → | BM25 | TF-IDF | BB2 | InL2 | IFB2 | LM |
|---|---|---|---|---|---|---|---|
| Base line | MAP | 0.4444 | 0.4455 | 0.3746 | 0.3909 | 0.3724 | 0.405 |
| | Rel.Ret. | 1683 | 1679 | 1659 | 1654 | 1664 | 1667 |
| YASS | MAP | 0.442 | 0.446 | 0.3773 | 0.3785 | 0.3675 | 0.3989 |
| | Rel.Ret. | 1764 | 1762 | 1729 | 1735 | 1731 | 1730 |
| FCB V-1 | MAP | 0.4476 | 0.4488 | 0.3767 | 0.3939 | 0.3746 | 0.4081 |
| | Rel.Ret. | 1725 | 1720 | 1703 | 1700 | 1708 | 1708 |
| SNS | MAP | 0.4483 | **0.4588** | **0.3798** | **0.3945** | **0.3808** | **0.4092** |
| | Rel.Ret. | 1745 | 1742 | 1732 | 1724 | 1728 | 1712 |
| GRAS | MAP | **0.4495** | 0.4505 | 0.3692 | 0.3815 | 0.3661 | 0.3995 |
| | Rel.Ret. | 1759 | 1755 | 1735 | 1741 | 1740 | 1736 |
| Trunc-n | MAP | 0.4543 | 0.4561 | 0.376 | 0.3879 | 0.3729 | 0.409 |
| | Rel.Ret. | 1749 | 1744 | 1733 | 1725 | 1735 | 1706 |

## 6 Evaluation

In the first set of experiments, we evaluate the effect of different language-independent stemming techniques in Indian languages IR. In Hindi, MAP, and the relevant documents retrieved in the baseline and different language-independent stemming methods are shown in Table 3. We conduct similar experiments for the other Indian languages: Gujarati and English, as shown in Table 4, and 5 respectively. It is observed that different stemming techniques improve MAP scores in Indian languages IR. The best performance by a stemming approach is shown in boldface. The SNS stemmer provides the best MAP score in Hindi and Gujarati. However, the GRAS stemmer provides the best MAP score in English. The trunc-n-based indexing strategy offers similar performance to the SNS stemmer in Hindi and Gujarati. Moreover, the GRAS stemmer provides comparable performance in English. During the evaluation of different retrieval models, we observed that the probabilistic retrieval models (BM25 and TF-IDF) give the best retrieval performance in Hindi, Gujarati and English. The DFR-based retrieval models (BB2, InL2, and IFB2) exhibit poor performance in Indian languages IR.

We perform a query-by-query analysis to get more insight into the effect of stemming in Indian languages. Here, we consider the best retrieval models and stemming approaches for Indian languages. We consider the SNS stemmer for Hindi and Gujarati languages and the GRAS stemmer for English. In closer observation, we found that stemming improves performance for 35 topics in Hindi and reduces performance for 15 topics. The performance of each query is shown in Fig. 1. Likewise, in Gujarati and English, stemming improves performance in 38, and 32 topics, respectively, and reduces performance in 8, and 12 topics. The percentage changes in performance due to stemming at the per-query level are shown in Fig 2, and 3. From query-by-query analysis, we also observed that the stemming performs better in Gujarati, and English than in Hindi.

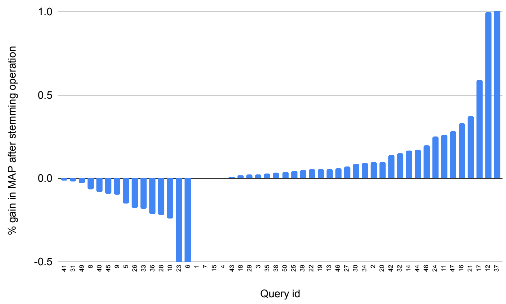



Figure 1: A query by query evaluation in Hindi by SNS stemmer in TF-IDF model

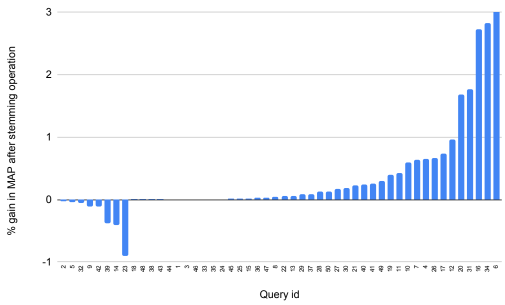



Figure 2: A query by query evaluation in Gujarati by SNS stemmer in InL2 model

During the experimentation of different

Table 4: Retrieval results in Gujarati 2011 text collection (50 T queries)

| | ↓ Parameter — R.M. → | BM25 | TF-IDF | BB2 | InL2 | IFB2 | LM |
|---|---|---|---|---|---|---|---|
| Base line | MAP | 0.24 | 0.2399 | 0.2041 | 0.1992 | 0.2021 | 0.2095 |
| | Rel.Ret. | 1315 | 1308 | 1278 | 1270 | 1274 | 1289 |
| YASS | MAP | 0.2464 | 0.2463 | 0.2116 | 0.2057 | 0.2077 | 0.2137 |
| | Rel.Ret. | 1320 | 1311 | 1280 | 1279 | 1276 | 1287 |
| FCB V-1 | MAP | 0.2423 | 0.2404 | 0.2056 | 0.1998 | 0.2031 | 0.2105 |
| | Rel.Ret. | 1337 | 1331 | 1292 | 1290 | 1296 | 1313 |
| SNS | MAP | **0.2647** | **0.2643** | **0.2385** | **0.2406** | **0.2342** | **0.2335** |
| | Rel.Ret. | 1359 | 1357 | 1343 | 1325 | 1336 | 1338 |
| GRAS | MAP | 0.2443 | 0.2439 | 0.2125 | 0.2167 | 0.2105 | 0.2184 |
| | Rel.Ret. | 1349 | 1342 | 1321 | 1305 | 1314 | 1329 |
| Trunc-n | MAP | 0.2579 | 0.2578 | 0.2282 | 0.2331 | 0.2246 | 0.2282 |
| | Rel.Ret. | 1360 | 1356 | 1342 | 1330 | 1333 | 1341 |

Table 5: Retrieval results in English 2011 text collection (50 T queries)

| | ↓ Parameter — R.M. → | BM25 | TF-IDF | BB2 | InL2 | IFB2 | LM |
|---|---|---|---|---|---|---|---|
| Base line | MAP | 0.2975 | 0.2981 | 0.2686 | 0.2633 | 0.2615 | 0.2543 |
| | Rel.Ret. | 2236 | 2232 | 2210 | 2204 | 2210 | 2182 |
| YASS | MAP | 0.3122 | 0.3133 | 0.2837 | 0.2769 | 0.2745 | 0.2652 |
| | Rel.Ret. | 2337 | 2337 | 2338 | 2312 | 2338 | 2278 |
| FCB V-1 | MAP | 0.3012 | 0.302 | 0.2723 | 0.2662 | 0.2651 | 0.257 |
| | Rel.Ret. | 2278 | 2277 | 2268 | 2249 | 2269 | 2221 |
| SNS | MAP | 0.3068 | 0.3065 | 0.2753 | 0.2709 | 0.2734 | 0.2611 |
| | Rel.Ret. | 2278 | 2279 | 2250 | 2240 | 2249 | 2224 |
| GRAS | MAP | **0.3145** | **0.3155** | **0.2849** | **0.2796** | **0.2763** | **0.2661** |
| | Rel.Ret. | 2310 | 2311 | 2294 | 2280 | 2290 | 2248 |
| Trunc-n | MAP | 0.3155 | 0.3164 | 0.2858 | 0.2818 | 0.2772 | 0.267 |
| | Rel.Ret. | 2309 | 2310 | 2295 | 2277 | 2290 | 2247 |

language-independent stemmers (shown in Table 3, 4, and 5), we observe that stemming improves retrieval performance in different Indian languages IR. On closer observation, we found that the effect of stemming varies in different Indian languages. The SNS stemmer performs best and improves a MAP score of 2.98% in Hindi, 20.78% in Gujarati IR. Similarly, the GRAS stemmer performs best and improves a MAP score of 5.83% in English IR. Among the different stemming techniques experimented with, the GRAS stemmer required less computational effort and performed best in different Indian languages. We conclude that the language-independent stemmer improves retrieval performance in different Indian languages IR. This observation is similar to the findings in other Indian and European languages by (Majumder et al., 2007) and (Paik and Parui, 2011).

## 7 Conclusion

Stemming is an essential preprocessing step in the IR system. The above experiments show that stemming improves retrieval performance in different Indian languages compared to the baseline approach (no stemming). Different stemming techniques perform best in Gujarati and English languages. However, the stemming technique provides a relatively poor performance in Hindi. The SNS stemmer performs best in Hindi and Gujarati, whereas the GRAS stemmer performs best in English. The trunc-n-based indexing strategy performs similarly to the best-stemming approaches in different Indian languages. During the evaluation of the retrieval models, we observe that the probabilistic retrieval models (BM25 and TF-IDF) perform best in Hindi, Gujarati and English languages. The DFR-based retrieval models provide poor performance in different Indian languages.

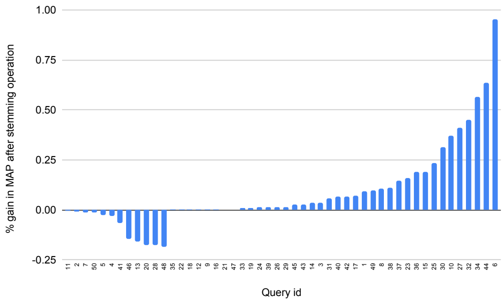



Figure 3: A query by query evaluation in English by GRAS stemmer in TF-IDF model

Although the effect of stemming is thoroughly investigated in the Indo-European language family, it is less explored in the Dravidian language family. India has significant native speakers in Dravidian languages such as Telugu, Tamil, Kannada, and Malayalam. So, it will be interesting to explore the effect of different stemming techniques in the Dravidian language family in the future. Moreover, one can also study the impact of different machine learning-based and deep learning-based stemming techniques in different Indian and European languages IR.

## 8 Acknowledgements

This work is supported by IIT (B.H.U), Varanasi. Moreover, the support and resources provided by the PARAM Shivay Facility under the National Supercomputing Mission, Government of India, at IIT (B.H.U), Varanasi, are gratefully acknowledged.

## References

Chris Buckley, Amit Singhal, Mandar Mitra, and Gerard Salton. 1995. New retrieval approaches using smart: Trec 4. In *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48.

John Dawson. 1974. Suffix removal and word conflation. *ALLC bulletin*, 2(3):33–46.

Ljiljana Dolamic and Jacques Savoy. 2010. Comparative study of indexing and search strategies for the hindi, marathi, and bengali languages. *ACM Transactions on Asian Language Information Processing (TALIP)*, 9(3):1–24.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational linguistics*, 27(2):153–198.

David A Hull. 1996. Stemming algorithms: A case study for detailed evaluation. *Journal of the American Society for Information Science*, 47(1):70–84.

Julie Beth Lovins. 1968. Development of a stemming algorithm. *Mech. Transl. Comput. Linguistics*, 11(1-2):22–31.

Prasenjit Majumder, Mandar Mitra, Swapan K Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. Yass: Yet another suffix stripper. *ACM transactions on information systems (TOIS)*, 25(4):18–es.

James Mayfield and Paul McNamee. 2003. Single n-gram stemming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 415–416.

Jiaul H Paik, Mandar Mitra, Swapan K Parui, and Kalervo Järvelin. 2011a. Gras: An effective and efficient stemming algorithm for information retrieval. *ACM Transactions on Information Systems (TOIS)*, 29(4):1–24.

Jiaul H Paik, Dipasree Pal, and Swapan K Parui. 2011b. A novel corpus-based stemming algorithm using co-occurrence statistics. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 863–872.

Jiaul H Paik and Swapan K Parui. 2011. A fast corpus-based stemmer. *ACM Transactions on Asian Language Information Processing (TALIP)*, 10(2):1–16.

Carol Peters. 2008. *Advances in Multilingual and Multimodal Information Retrieval: 8th Workshop of the Cross-Language Evaluation Forum, CLEF 2007, Budapest, Hungary, September 19-21, 2007, Revised Selected Papers*, volume 5152. Springer Science & Business Media.

Martin F Porter. 1980. An algorithm for suffix stripping. *Program*.

Siba Sankar Sahu, Debrup Dutta, Sukomal Pal, and Imran Rasheed. 2023. Effect of stopwords and stemming techniques in urdu ir. *SN Computer Science*, 4(5):547.

Siba Sankar Sahu and Sukomal Pal. 2023. Building a text retrieval system for the sanskrit language: Exploring indexing, stemming, and searching issues. *Computer Speech & Language*, 81:101518.

Jacques Savoy. 2006. Light stemming approaches for the french, portuguese, german and hungarian languages. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1031–1035.

Gianmaria Silvello, Riccardo Bucco, Giulio Busato, Giacomo Fornari, Andrea Langeli, Alberto Purpura, Giacomo Rocco, Alessandro Tezza, and Maristella Agosti. 2018. Statistical stemmers: A reproducibility study. In *European Conference on Information Retrieval*, pages 385–397. Springer.

Jinxi Xu and W Bruce Croft. 1998. Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems (TOIS)*, 16(1):61–81.

# Checklist Engineering Empowers Multilingual LLM Judges

**Mohammad Ghiasvand Mohammadkhani**
Amirkabir University of Technology
mohammad.ghiasvand@aut.ac.ir

**Hamid Beigy**
Sharif University of Technology
beigy@sharif.edu

## Abstract

Automated text evaluation has long been a central issue in Natural Language Processing (NLP). Recently, the field has shifted toward using Large Language Models (LLMs) as evaluators—a trend known as the LLM-as-a-Judge paradigm. While promising and easily adaptable across tasks, this approach has seen limited exploration in multilingual contexts. Existing multilingual studies often rely on proprietary models or require extensive training data for fine-tuning, raising concerns about cost, time, and efficiency. In this paper, we propose ***C**hecklist **E**ngineering based LLM-as-a-**Judge*** (***CE-Judge***), a training-free framework that uses checklist intuition for multilingual evaluation with an open-source model. Experiments across multiple languages and three benchmark datasets, under both pointwise and pairwise settings, show that our method generally surpasses the baselines and performs on par with the GPT-4o model.[1]

## 1 Introduction

Evaluation is a fundamental task in Natural Language Processing (NLP) for measuring a model's performance on specific tasks. Automating this process offers significant benefits and has been a focus since the early stages of NLP research. Moreover, beyond creating evaluators proficient in English, it is crucial to develop their evaluation capabilities in parallel for other languages. Traditional evaluation metrics (Papineni et al., 2002) have some drawbacks, such as the necessity of reference answers and a lack of interpretability, which has led to a paradigm shift toward developing Large Language Model (LLM) evaluators, referred to as LLM-as-a-Judge (Gu et al., 2025; Li et al., 2024). These models are also capable of evaluating long-form LLM generations in either a pointwise or pairwise format—meaning grading a single response or selecting the better response out of two, respectively. Some advantages of this approach include high adaptability (Bavaresco et al.) and interpretability, in contrast to traditional metrics, as well as low inference time and the fact that the evaluated LLM does not need to be active during evaluation (i.e., it does not need to generate additional responses), both in contrast to more complex LLM-based evaluation frameworks such as Kim et al. (2025).

Despite significant efforts to make LLMs multilingual (Qin et al., 2024), extending LLM-as-a-Judge to multilingual configurations has received relatively little attention. Although current multilingual LLM judges (Pombal et al., 2025; Doddapaneni et al., 2025) perform well, their main limitation is their reliance on proprietary models or the need for a large amount of real or synthetic data to fine-tune a capable evaluator, raising concerns regarding cost, time, and efficiency.

Meanwhile, checklists as interpretable evaluation tools (Doddapaneni et al., 2024; Cook et al.) are gaining traction for their transparency and structure, although their application to multilingual evaluation remains relatively underexplored, and most of them also lack robust support for pairwise evaluation. For instance, (Wei et al., 2025) suggests to handle the pairwise setting by selecting the higher-graded response based on independent pointwise scores, which fails to capture the nuanced comparative superiority between responses. In this work, we present CE-Judge, an LLM-as-a-Judge framework that builds and uses engineered checklists for evaluation. It supports multilingual evaluation and both pointwise and pairwise modes. Our pipeline follows a three-stage process: the first two stages aim to generate broad and dynamic checklist items, and the third applies them for judgment. Notably, by using a lightweight, open-source LLM without any fine-tuning, our method demonstrates strong performance across different evaluation scenarios.

[1]The code implementation is accessible at https://github.com/mghiasvand1/CE-Judge.



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 190–196, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-021

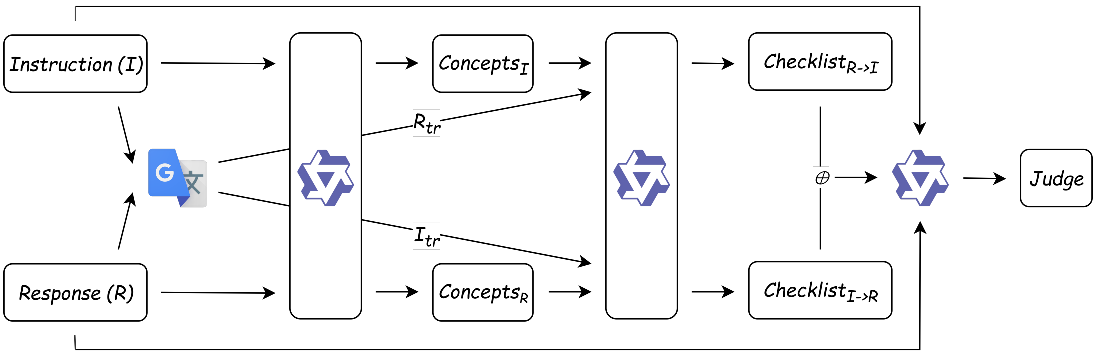



Figure 1: CE-Judge framework illustration.

## 2 Related Works

### 2.1 LLM-as-a-Judge

We begin with Zheng et al. (2023), which uses the generative capabilities of LLMs to act as evaluators. These can be grouped into two types: prompt-based and fine-tuned evaluators. For instance, for the former, Li et al. (2025) adopts a "decompose and aggregate" strategy, identifying weighted evaluation aspects and combining their scores to assess candidate responses. For the latter, which has recently gained momentum, Kim et al. (2024) is a representative work that trains evaluators using large-scale synthetic data in both pointwise and pairwise setups, incorporating a weight-merging technique. In multilingual settings, consistency remains limited, as noted by Fu and Liu (2025). Among the few multilingual methods, Doddapaneni et al. (2025) translates between languages to anchor outputs in English for consistent scoring, while Pombal et al. (2025) follows Kim et al. (2024) to generate multilingual evaluation data and fine-tune models accordingly. Chang et al. (2025) investigates several aspects of multilingual LLM-based evaluators, including reference-free prompting, the effect of language resource availability, and the impact of fine-tuning. Thellmann et al. (2024) creates various multilingual evaluation benchmarks while exploring the impact of translation and evaluating LLMs.

### 2.2 Checklist-based Evaluators

Several works have explored checklist-based evaluation. RocketEval (Wei et al., 2025) generates binary checklist items, then reweights them to produce final scores. TICK (Cook et al.) uses instructions to generate checklists, which LLMs use for self-improvement. CheckEval (Lee et al., 2024) defines high-level criteria, then decomposes, diversifies, and filters them to form evaluation checklists. FBI (Doddapaneni et al., 2024) employs checklists for meta-evaluation to assess evaluator LLMs. Unlike these works, our framework introduces a novel architecture and, notably: (1) extends to multilingual settings; (2) supports pairwise evaluation beyond pointwise framing; and (3) uniquely incorporates broadness, descriptiveness, dynamism, and answer-mentioning in a unified manner.

## 3 CE-Judge Pipeline

We present our training-free, efficient evaluation framework (Figure 1), which consists of three steps. The pipeline aims, for each case, to first construct an engineered checklist, followed by utilizing this checklist to enhance the decisions of the evaluator LLM. All LLM generations are asked to be in English to leverage its strong performance in the language (Mondshine et al., 2025). Our framework, within its architecture, targets the development of a level-by-level multilingual understanding of input-output pair evaluation, integrating input-output linkages to enable dynamism while considering the breadth of the checklist, allowing the evaluator LLM to identify relevant criteria within its context and make judgments based on its decisions.

### 3.1 Concepts Generation

Considering an instruction as input—replaced by the source text in the translation evaluation task—and a corresponding response, we pass each separately to the LLM along with the prompt in 4.4.1 for concept generation. This generation aims to produce an abstract-level text that represents the skeleton of the corresponding text.

### 3.2 Checklist Generation

Next, we translate both the instruction and response texts into English. The reason for the translation is

| Model | MMEval (Reasoning) | | | | | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | en | de | fr | es | ru | zh | bn | ja | th | te | sw | |
| **Proprietary Models** | | | | | | | | | | | | |
| GPT-4o | **0.79** | 0.79 | **0.78** | **0.79** | 0.76 | **0.78** | **0.84** | 0.80 | **0.79** | **0.87** | **0.80** | **0.79** |
| **Medium (7B parameters)** | | | | | | | | | | | | |
| Qwen2.5-7B-Instruct | 0.67 | 0.65 | 0.63 | 0.65 | 0.66 | 0.71 | 0.60 | 0.62 | 0.66 | 0.67 | 0.61 | 0.64 |
| Hercule 7B | 0.50 | 0.56 | 0.55 | 0.55 | 0.53 | 0.57 | 0.57 | 0.54 | 0.52 | 0.54 | 0.51 | 0.54 |
| M-Prometheus 7B | 0.60 | 0.62 | 0.63 | 0.60 | 0.62 | 0.69 | 0.61 | 0.57 | 0.60 | 0.65 | 0.72 | 0.62 |
| **Large (14B+ parameters)** | | | | | | | | | | | | |
| Prometheus 2 8x7B | 0.54 | 0.65 | 0.58 | 0.58 | 0.58 | 0.64 | 0.57 | 0.56 | 0.60 | 0.60 | 0.63 | 0.59 |
| M-Prometheus 14B | 0.64 | 0.70 | 0.70 | 0.69 | 0.69 | 0.72 | 0.70 | 0.70 | 0.68 | 0.72 | 0.76 | 0.70 |
| **Ours (7B parameters)** | | | | | | | | | | | | |
| CE-Judge | 0.77 | **0.81** | 0.77 | 0.72 | **0.78** | 0.77 | 0.75 | **0.84** | 0.78 | 0.76 | 0.78 | 0.77 |

Table 1: Accuracy on MMEval (Reasoning) broken down by language.

to ensure that either the instruction or the response is in the same language as the previously generated concepts. Using the translated response, the concepts generated from the instruction (from the previous step), and the prompt in 4.4.2, we generate a checklist following the "response to instruction" direction. Following this direction means formulating questions about criteria that are not specified in the instruction's concepts but are suggested by the response. Likewise, we use the translated instruction and the response's concepts to generate a checklist for the "instruction to response" direction, which points to the evaluation criteria suggested by the instruction. This dual approach aims to blind each side once, broadening checklist coverage and enhancing awareness of both sides' content, rather than relying on a standard checklist with limited, predefined criteria. In this step, we also avoid prejudgment and ask the model to generate more descriptive items, going beyond simple binary questions.

### 3.3 Judgment

The final step is judgment. First, the two checklists from the previous steps are concatenated into a unified checklist. It's important to note that the entire process described so far is for pointwise evaluation. For pairwise evaluation, the process remains the same, except that the previous two steps are applied to two candidate responses instead of one. As a result, after concatenation, we obtain two checklists, one for each candidate. In this step, we provide the untranslated versions of the instruction and response(s), along with the checklist(s) and the prompt template in 4.4.3. We used each instruction or response in its original language to avoid the negative effects of translation biases, because this step—unlike the previous step, which was an intermediate step for generating checklist items—is the final step, and having access to accurate real data is crucial. The LLM is then asked to answer a subset of key checklist items and generate evaluation feedback. Unlike prior works where checklist items are marked with ticks, crosses, or weighted scores, here the model exercises discretion in its judgments, and the final evaluation is left to the model's decision.

## 4 Experiments

### 4.1 Experimental Setup

In this work, we used the `Qwen2.5-7B-Instruct` model (Yang et al., 2024) as the backbone LLM, accessed freely via the *Novita API*[2]. The hyperparameters "temperature", "top_p", and "seed" were set to 0, 1, and 42, respectively, to ensure reproducibility. For translation, we employed the free *Google Translate API* available through the `deep-translator` Python package[3].

[2] https://novita.ai/
[3] https://deep-translator.readthedocs.io/en/latest/README.html

| Model | MMEval (Chat) | | | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|
| | en | de | fr | es | ca | ru | zh | |
| **Proprietary Models** | | | | | | | | |
| GPT-4o | **0.72** | 0.70 | **0.73** | 0.64 | 0.75 | **0.78** | 0.80 | <u>0.73</u> |
| **Medium (7B parameters)** | | | | | | | | |
| Qwen2.5-7B-Instruct | <u>0.69</u> | **0.75** | 0.71 | **0.78** | 0.72 | 0.66 | <u>0.85</u> | 0.72 |
| Hercule 7B | 0.62 | 0.71 | 0.61 | 0.55 | 0.62 | 0.64 | 0.65 | 0.62 |
| M-Prometheus 7B | 0.68 | 0.65 | 0.66 | 0.59 | 0.62 | 0.56 | 0.58 | 0.62 |
| **Large (14B+ parameters)** | | | | | | | | |
| Prometheus 2 8x7B | 0.64 | 0.68 | <u>0.72</u> | 0.65 | <u>0.77</u> | 0.64 | 0.80 | 0.70 |
| M-Prometheus 14B | 0.61 | <u>0.72</u> | 0.64 | 0.64 | 0.57 | 0.71 | 0.73 | 0.66 |
| **Ours (7B parameters)** | | | | | | | | |
| CE-Judge | <u>0.69</u> | 0.60 | **0.73** | <u>0.77</u> | **0.82** | <u>0.77</u> | **0.87** | **0.75** |

Table 2: Accuracy on MMEval (Chat) broken down by language.

### 4.2 Datasets

Since our method is training-free, all datasets are used solely for testing. We evaluated our framework in both pointwise and pairwise settings. For pointwise evaluation, we used the student-annotated subset of the LitEval (Zhang et al., 2024), which contains source–target literary translations for four language pairs with human ratings from 1 to 7. For pairwise evaluation, we employed the reasoning and chat subsets of the MM-Eval dataset (Son et al., 2024), covering 11 and 7 languages, respectively. Each input consists of a reasoning question or chat history, with the task being to choose the better of two candidate responses. The reason for utilizing the LitEval and MM-Eval datasets is that the former is one of the only multilingual pointwise evaluation datasets, and the latter is more robust than the well-known M-RewardBench multilingual benchmark (Gureja et al., 2025).

### 4.3 Evaluation Metrics

To evaluate our CE-Judge framework in pointwise mode, we measured performance using Kendall's Tau correlation coefficient (Kendall, 1938), which assesses agreement between our model's rankings and human judgments. For the pairwise setting, we used accuracy—defined as the number of correct predictions over the total number of samples.

### 4.4 Prompt Templates

In this section, we list all the prompts used within our framework.

#### 4.4.1 Concepts Generation Prompts

The prompts for this step, across all three datasets, are shown in figure 2, and the "[INPUT]" placeholder must be replaced with the text from which we want to extract concepts, such as an instruction, response, etc.

#### 4.4.2 Checklist Generation Prompts

Figures 3, 4, and 5 show checklist generation prompts for Liteval, MM-Eval (Reasoning), and MM-Eval (Chat), respectively. Each figure consists of two prompts indicating the checklist creation direction. Note that the "[CONCEPTS]" placeholder must be replaced with the concepts generated in the previous step.

#### 4.4.3 Judgment Prompts

We only use system prompts from this section, which are shown in figure 6: one for the Liteval dataset and another for the MM-Eval datasets. Figure 7 presents the prompt template for the Liteval dataset, while figure 8 shows the prompts for the two MM-Eval datasets. In these prompts, the placeholders clearly indicate what should replace them. Importantly, to demonstrate the flexibility of our framework, we also use a scoring guide for the pointwise assessment to help our judge LLM perform a more accurate evaluation.

### 4.5 Baselines

We compare our framework with three types of models. The first includes proprietary models like `GPT-4o`. The second is `Qwen2.5-7B-Instruct`,

| Model | LitEval | | | | Avg. |
|---|---|---|---|---|---|
| | de→en | en→de | en→zh | de→zh | |
| **Proprietary Models** | | | | | |
| GPT-4o | 0.26 | 0.48 | 0.41 | 0.40 | 0.38 |
| **Medium (7B parameters)** | | | | | |
| Qwen2.5-7B-Instruct | 0.12 | 0.32 | 0.17 | 0.07 | 0.17 |
| Hercule 7B | 0.26 | 0.33 | 0.38 | 0.42 | 0.34 |
| M-Prometheus 7B | 0.20 | <u>0.53</u> | 0.46 | <u>0.54</u> | <u>0.43</u> |
| **Large (14B+ parameters)** | | | | | |
| Prometheus 2 8x7B | 0.24 | 0.36 | 0.25 | 0.40 | 0.31 |
| M-Prometheus 14B | **0.29** | **0.57** | <u>0.48</u> | **0.56** | **0.47** |
| **Ours (7B parameters)** | | | | | |
| CE-Judge | <u>0.28</u> | 0.46 | **0.49** | 0.30 | 0.38 |

Table 3: Kendall correlation on LitEval broken down by language pair.

a strong multilingual open-source LLM that is instruction-tuned from a pretrained model without further fine-tuning. The third category consists of models explicitly trained as evaluators, such as `Prometheus 2` (Kim et al., 2024), `Hercule` (Doddapaneni et al., 2025), and `M-Prometheus` (Pombal et al., 2025), as discussed in Subsection 2.1.

## 5 Results

We evaluate CE-Judge on three multilingual evaluation datasets—reasoning, chat, and literary translation—against proprietary and open-source baselines, including the fine-tuned M-Prometheus. In all three tables, languages are shown by their codes, and, more importantly, the results for the other models are taken from Pombal et al. (2025).

- In the reasoning evaluation task (Table 1), CE-Judge achieves an average accuracy of **0.77**, outperforming all open-source baselines in all languages, including large fine-tuned evaluators such as M-Prometheus 14B. Despite being training-free and based on a 7B-parameter model, it performs competitively with GPT-4o (which has an average accuracy of 0.79) and maintains strong performance across both high- and low-resource languages.

- In the chat evaluation (Table 2), CE-Judge achieves an average accuracy of **0.75**, surpassing GPT-4o (with the average of 0.73) and significantly outperforming the M-Prometheus models across nearly all languages. This result highlights the robustness of our checklist-driven approach in conversational scenarios that require nuanced, context-aware judgment.

- In the literary translation evaluation (Table 3), which requires nuanced linguistic and stylistic understanding, CE-Judge achieves an average Kendall's Tau correlation of **0.38**, significantly outperforming its backbone model, Qwen2.5-7B, and delivering performance comparable to GPT-4o. Although it slightly lags behind M-Prometheus 7B (average of 0.43)—which benefits from fine-tuning on supervised machine translation evaluation data—our training-free approach remains highly competitive.

## 6 Conclusion

In this work, we introduce CE-Judge, a novel and straightforward checklist-based framework for multilingual LLM-as-a-Judge that is training-free and built on an open-source model. By leveraging dynamic, broad, and flexible checklist items, CE-Judge supports both pointwise and pairwise evaluations across diverse languages. Experiments on multiple multilingual benchmarks show that CE-Judge not only generally outperforms open-source fine-tuned baselines but also performs on par with GPT-4o. These results highlight the promise of structured, dynamic evaluation techniques for improving the reliability and interpretability of LLM judgment, particularly in multilingual contexts, for more consistent performance.

## Ethics Statement

This study aims to advance multilingual evaluation using a training-free approach built on an open-source LLM, prioritizing accessibility and transparency. We leveraged publicly available datasets and APIs, with no collection of personal or sensitive data. All experiments are free from human involvement and pose no privacy or safety risks.

## Limitations

Despite its strong results and training-free design, our framework has several limitations that should be addressed in future work. First, for our concept and checklist generation steps, it is worthwhile to try few-shot learning to ensure the numbered points in the task description of the prompts are applied accurately. Second, it is important to evaluate our method more extensively beyond the three tasks discussed, which could be facilitated by an automatic prompt generation module that creates step-specific prompts and removes the need for manual design. Third, our method relies solely on LLM generation, which may suffer from misalignment between training objectives and robust text generation. Incorporating internal LLM representations, as shown by Sheng et al. (2024), could capture more accurate implicit knowledge. Finally, our framework's flexibility suggests potential extensions as a plug-and-play method or adaptations to other evaluation strategies, such as interview-based evaluation (Kim et al., 2025).

## References


Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, and 1 others. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *arXiv preprint arXiv:2406.18403*.

Jiayi Chang, Mingqi Gao, Xinyu Hu, and Xiaojun Wan. 2025. Exploring the multilingual nlg evaluation abilities of llm-based evaluators. *Preprint*, arXiv:2503.04360.

Jonathan Cook, Tim Rocktäschel, Jakob Nicolaus Foerster, Dennis Aumiller, and Alex Wang. Ticking all the boxes: Generated checklists improve llm evaluation and generation. In *Language Gamification-NeurIPS 2024 Workshop*.

Sumanth Doddapaneni, Mohammed Khan, Sshubam Verma, and Mitesh M Khapra. 2024. Finding blind spots in evaluator llms with interpretable checklists. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16279–16309.

Sumanth Doddapaneni, Mohammed Safi Ur Rahman Khan, Dilip Venkatesh, Raj Dabre, Anoop Kunchukuttan, and Mitesh M Khapra. 2025. Cross-lingual auto evaluation for assessing multilingual LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29297–29329, Vienna, Austria. Association for Computational Linguistics.

Xiyan Fu and Wei Liu. 2025. How reliable is multilingual llm-as-a-judge? *arXiv preprint arXiv:2505.12201*.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Srishti Gureja, Lester James Validad Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Triandi Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. 2025. M-RewardBench: Evaluating reward models in multilingual settings. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 43–58, Vienna, Austria. Association for Computational Linguistics.

Maurice G Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93.

Eunsu Kim, Juyoung Suk, Seungone Kim, Niklas Muennighoff, Dongkwan Kim, and Alice Oh. 2025. LLM-as-an-interviewer: Beyond static testing through dynamic LLM evaluation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 26456–26493, Vienna, Austria. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Yukyung Lee, Joonghoon Kim, Jaehee Kim, Hyowon Cho, and Pilsung Kang. 2024. Checkeval: Robust evaluation framework using large language model via checklist. *arXiv preprint arXiv:2403.18771*.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2024. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Minzhi Li, Zhengyuan Liu, Shumin Deng, Shafiq Joty, Nancy Chen, and Min-Yen Kan. 2025. Dna-eval: Enhancing large language model evaluation through decomposition and aggregation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2277–2290.

Itai Mondshine, Tzuf Paz-Argaman, and Reut Tsarfaty. 2025. Beyond English: The impact of prompt translation strategies across languages and tasks in multilingual LLMs. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1331–1354, Albuquerque, New Mexico. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

José Pombal, Dongkeun Yoon, Patrick Fernandes, Ian Wu, Seungone Kim, Ricardo Rei, Graham Neubig, and André F. T. Martins. 2025. M-prometheus: A suite of open multilingual llm judges. *Preprint*, arXiv:2504.04953.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*.

Shuqian Sheng, Yi Xu, Tianhang Zhang, Zanwei Shen, Luoyi Fu, Jiaxin Ding, Lei Zhou, Xiaoying Gan, Xinbing Wang, and Chenghu Zhou. 2024. Repeval: Effective text evaluation with llm representation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7019–7033.

Guijin Son, Dongkeun Yoon, Juyoung Suk, Javier Aula-Blasco, Mano Aslan, Vu Trong Kim, Shayekh Bin Islam, Jaume Prats-Cristià, Lucía Tormo-Bañuelos, and Seungone Kim. 2024. Mm-eval: A multilingual meta-evaluation benchmark for llm-as-a-judge and reward models. *arXiv preprint arXiv:2410.17578*.

Klaudia Thellmann, Bernhard Stadler, Michael Fromm, Jasper Schulze Buschhoff, Alex Jude, Fabio Barth, Johannes Leveling, Nicolas Flores-Herr, Joachim Köhler, René Jäkel, and Mehdi Ali. 2024. Towards multilingual llm evaluation for european languages. *Preprint*, arXiv:2410.08928.

Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. 2025. Rocketeval: Efficient automated LLM evaluation via grading checklist. In *The Thirteenth International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Ran Zhang, Wei Zhao, and Steffen Eger. 2024. How good are llms for literary translation, really? literary translation evaluation with humans and llms. *arXiv preprint arXiv:2410.18697*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

# C A N C E R: Corpus for Accurate Non-English Cancer-related Educational Resources

**Anika Harju**[1] **Asma Shakeel**[2] **Tiantian He**[3] **Tianqi Xu**[3] **Aaro Harju**[4]

[1] University of Technology Sydney, Australia

[2] School of Electrical Engineering and Computer Science, NUST, Pakistan

[3] IT University of Copenhagen, Denmark

[4] Independent Researcher

Anika.Harju@.uts.edu.au, ashakeel.msee18seecs@seecs.edu.pk,
{tihe,tixu}@itu.dk, aaro.m.harju@outlook.com

## Abstract

Improving the quality of cancer terminology through Machine Translation (MT) in non-English languages remains an under-researched area despite its critical role in supporting self-management and advancing multilingual patient education. Existing computational tools encounter significant limitations in accurately translating cancer terminologies, particularly for low-resource languages, primarily due to data scarcity and morphological complexity. To address the gap, we introduce a dedicated terminology resource —Corpus for Accurate Non-English Cancer-related Educational Resources (C A N C E R), a manually annotated dataset in Finnish (FI), Chinese (ZH), and Urdu (UR), curated from publicly available existing English (EN) data. We also examine the impact of data quality versus quantity and compare the performance of the Opus-mt-en-fi, Opus-mt-en-zh, and Opus-mt-en-ur models with the SMaLL-100 multilingual MT model. We assess translation quality using automatic and human evaluation. Results demonstrated that high-quality parallel data, though sparse, combined with fine-tuning, substantially improved the translation of cancer terminology across both high and low-resource language pairs, positioning the C A N C E R corpus as a foundational resource for improving multilingual patient education.[1]

## 1 Introduction

Cancer remains a major global health challenge, representing one of the leading causes of death worldwide (Bray et al., 2021). Patient education is critical for understanding the cancer diagnosis and undergoing the intensive treatment (Cai et al., 2023). There is a significant demand for simplifying complex cancer terminology through Machine Translation (MT) in patient education materials to improve health literacy (Oniani et al., 2023). The persistent research gap impedes effective cancer patient education and increases the risk of misdiagnosis and adverse outcomes (Kasperė et al., 2023). Moreover, the World Health Organization International Classification of Diseases recommends the translation of medical terminology into other languages to enhance accessibility, as codes and classifications containing the ontologies are primarily in English (EN) (Harrison et al., 2021). Consequently, the accurate translation of medical terminology, particularly for diseases such as cancer, is critical for advancing cancer patient education and self-management (McCorkle et al., 2011) in support of patients with limited proficiency in the native language where they reside (Castilla et al., 2005; Lovis et al., 1998).

Despite the high proficiency of state-of-the-art (SOTA) Neural Machine Translation (NMT) models (Dabre et al., 2020; Wang et al., 2023), MT of medical terminology has fallen short (Nayak et al., 2023). Even with various fine-tuning approaches, NMT models still struggle to translate medical terminology accurately (Nayak et al., 2020). One approach to mitigate the issue is to utilize a high-quality parallel dataset for MT training (de Gibert Bonet et al., 2022). However, annotated parallel medical data remain scarce — particularly in the cancer domain (Ma et al., 2020). Furthermore, the computational demands associated with implementing MT on SOTA models are costly (Nayak et al., 2023; Park et al., 2021; Zhang et al., 2023).

In this paper, we focus on fine-tuning three NMT models (Opus-mt-en-fi, Opus-mt-en-zh, and Opus-mt-en-ur) (Tiedemann and de Gibert, 2023) and a multilingual MT model (SMaLL-100) (Mohammadshahi et al., 2022) using manually annotated training data derived from EN segments of the public English-Chinese Cancer Parallel Corpus (ECCParaCorp) (Ma et al., 2020) to construct new EN-to-FI and EN-to-UR parallel corpora and extend language coverage of the existing EN-to-

[1] Available at: C A N C E R Corpus



*Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025*, pages 197–209, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-022

| Annotated data | # Pairs |
|---|---|
| **In-domain** | |
| EN-to-FI | 1,494 |
| EN-to-ZH | 1,494 |
| EN-to-UR | 1,494 |
| **Out-of-domain** | |
| EN-to-FI | 291 |
| EN-to-ZH | 291 |
| EN-to-UR | 291 |
| Total | 5,355 |

Table 1: Annotated cancer terminology parallel data

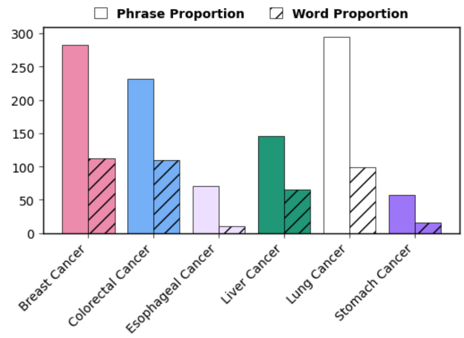



Figure 1: Categories of terminology data in cancer advocacy colors

ZH language pair. We assess translation quality using automatic evaluation metrics (Papineni et al., 2002; Popović, 2015; Rei et al., 2020) and human evaluation (Escribe, 2019). We also evaluate generalization using human evaluation on three manually annotated parallel datasets (EN–FI, EN–ZH, and EN–UR) curated from the public glossary on the Peter MacCallum Cancer Centre website (MacCallum, 2024).

Our paper focuses on improving the translation quality of cancer terminologies in two high-resource languages (FI and ZH) and a low-resource language (UR) to advance cancer patient education and bridge language challenges to support improved self-management (Lovis et al., 1998).

Our contributions can be summarized as follows:

- Creation of C A N C E R, a manually annotated corpus, to advance cancer patient education and self-management in EN-to-FI, EN-to-ZH, and EN-to-UR language pairs.
- Adaptation of the Opus-mt-en-fi, Opus-mt-en-zh, Opus-mt-en-ur, and the SMaLL-100 multilingual MT model, through fine-tuning to improve the translation quality of cancer terminologies.
- In-depth analysis of automatic performance metrics, including human evaluation by medical practitioners and native FI, ZH, and UR speakers provided insights into the translation quality of the cancer terminologies.

## 2 Data

In the first data acquisition step, we collected EN data from Ma et al. (2020) cancer corpus, which includes cancer terminologies (411 words and 1,083 phrases) (Table 1) related to cancer prevention, screening, diagnosis, and treatment. Using the existing EN source data, we manually annotated the FI and UR references to create two parallel datasets (EN-to-FI and EN-to-UR) while extending language coverage with the EN-to-ZH pair for the training, development, and test splits (Ma et al., 2020). We excluded sentence-level data to focus exclusively on terminology-level translation. The C A N C E R corpus includes data in six categories: Breast Cancer, Colorectal Cancer, Esophageal Cancer, Liver Cancer, Lung Cancer, and Stomach Cancer (Figure 1). In the second step, we compiled EN data from MacCallum (2024) online glossary that covers commonly used cancer-related terminologies (182 words and 109 phrases) (Table 1) from A-to-Z during diagnosis and treatment, and manually annotated FI, ZH, and UR references to the source data to create three out-of-domain datasets (EN-to-FI, EN-to-ZH, and EN-to-UR) to assess generalization using human evaluation (Escribe, 2019). We annotated our five parallel cancer terminology datasets, leveraging the expertise of medical practitioners and native FI, ZH, and UR speakers, effectively addressing the data scarcity gap (Lovis et al., 1998).

## 3 Related Work

Translating medical terminologies is a challenging task. The unique features of different languages, combined with the complexity of medical jargon and data scarcity, have further hindered efforts. (Ao and Acharya, 2021) Moreover, medical institutions have limited specialized health educators to support self-healthcare in chronic diseases such as cancer, particularly for non-native English speakers (Ugas et al., 2024). Existing studies have sought to improve translation quality across the medical domain, including work to enhance medical terminology, without specific emphasis on the cancer field (Alam et al., 2021). Prior research included exploring the time frame required to translate newly introduced or revised medical terminology for evaluation by healthcare experts (Skianis et al., 2020). The Castilla et al. (2005) study also investigated automated evaluation of medical terminologies using the Unified Medical Language System to assess cross-lingual information clinical data extracted from Portuguese-language thoracic radiology reports. During human evaluation, the Kasperė et al. (2024) study, however, found that the translation of medical terminology from English to Lithuanian was of poor quality, concluding that MT should serve as a supplementary approach only. In contrast, the Herrera-Espejel and Rach (2023) study highlighted MT as a potential solution to bridge language barriers in public health communication that restrict access to essential information for culturally and linguistically diverse groups.

In our experiment, we rely on manually annotated domain-specific data (Table 1) and fine-tuning techniques to adapt the Opus-MT models and the SMaLL-100 multilingual MT model to the unique morphological characteristics of FI, ZH, and UR cancer terminologies to advance health education (Oniani et al., 2023), and support patients to overcome language barriers, particularly when taking prescribed medication and navigating digital platforms (Lorig and Holman, 2003; McCorkle et al., 2011).

## 4 Method

We denote, let $X = \{x_1, x_2, \ldots, x_N\}$ as the source language (EN) consisting of $N$ medical terminologies. $Y = \{y_1, y_2, \ldots, y_N\}$ as its corresponding target-language (FI, ZH and UR). Each pair $(x_i, y_i)$ constitutes a parallel medical terminology. The probability of translating the entire target sequence $Y$ given the source sequence $X$ can be approximated as:

$$P(Y \mid X; \theta) \approx \prod_{i=1}^{N} P(y_i \mid x_i; \theta)$$

**OPUS-MT** In the first stage of the experiment, we fine-tuned the Opus-MT models on annotated parallel training data (EN-FI, EN-ZH, and EN-UR). We utilized dynamic batching with the Hugging Face DataCollatorForSeq2Seq (Solanki and Khublani, 2024) and systematically optimized hyperparameters by experimenting with batch sizes 8, 16, and 32 (achieving the best performance with a batch size of 8) and a learning rate grid search (optimal rate: 6e-04) over three epochs (Appendix A). Label smoothing (probability = 0.1) was applied to enhance precision. We evaluated model performance using the bilingual evaluation understudy (BLEU) (Vaswani, 2017), CHaRacter-level F-score (CHRF) (Popović, 2015), and Cross-lingual Optimized Metric for Evaluation of Translation (COMET) (Rei et al., 2020) metrics .

**SMaLL-100** The second stage of the experiment involved prepending the EN language token (_en_) to the encoder input in the SMaLL-100 model to specify the source language explicitly. To prompt the decoder to generate translations in the correct target language, we added a beginning-of-sequence (BOS) token via the forced_bos_token_id parameter. We applied similar hyperparameter settings (Appendix A) as in the first experiment to ensure consistency across model comparisons, using an optimal learning rate of 7e-05 (achieving the best performance with a batch size of 8) (Fuady et al., 2024). Native speakers assessed the generated translations on the in-domain test data from both experiments. To evaluate generalization, we selected the models with the lowest validation loss and assessed translation quality on out-of-domain datasets using human evaluation (Escribe, 2019).

## 5 Results

**OPUS-MT** The models demonstrated varying levels of translation effectiveness across the EN-FI, EN-ZH, and EN-UR language pairs. The Opus-mt-en-fi model achieved the highest BLEU score (Table 2), suggesting robust translation quality. CHRF and COMET scores (Appendices B & C) were also consistently high, indicating strong alignment with reference translations at the character and semantic level. The stability highlighted the capacity of the Opus-mt-en-fi model to adapt to the intricate morphological structure of the FI language, reinforcing its suitability for the MT task. Similarly, the Opus-mt-en-zh model exhibited satisfactory performance across various configurations (Appendices B & C), highlighting the ability to understand the language patterns. However, performance dipped with the Opus-mt-en-ur model, as challenges persist in generalizing across

| Batch Size | Opus-mt-en-fi | | | Opus-mt-en-zh | | | Opus-mt-en-ur | | | EN-FI | | | EN-ZH | | | EN-UR | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | SMaLL-100 Multilingual MT Model | | | | | | | | |
| | BLEU | CHRF | COMET | BLEU | CHRF | COMET | BLEU | CHRF | COMET | BLEU | CHRF | COMET | BLEU | CHRF | COMET | BLEU | CHRF | COMET |
| **Baseline** | | | | | | | | | | | | | | | | | | |
| 8 | 12.95 | 51.12 | 82.06 | 7.61 | 24.14 | 75.77 | 2.38 | 16.51 | 51.26 | 3.43 | 13.82 | 65.30 | 2.78 | 6.38 | 65.62 | 2.60 | 1.68 | 54.87 |
| 16 | 12.73 | 50.58 | 81.75 | 3.67 | 21.28 | 75.02 | 2.17 | 16.46 | 51.16 | 2.40 | 13.39 | 64.48 | 1.43 | 5.41 | 64.50 | 1.34 | 1.53 | 53.83 |
| 32 | 11.52 | 49.81 | 81.07 | 2.38 | 19.05 | 73.89 | 2.08 | 16.44 | 51.12 | 1.92 | 12.81 | 63.51 | 1.30 | 5.24 | 64.11 | 1.22 | 1.51 | 53.45 |
| **Fine-tuned** | | | | | | | | | | | | | | | | | | |
| **8** | **58.25** | **75.22** | **92.24** | **41.28** | **48.46** | **86.96** | **28.60** | **47.20** | **68.30** | **54.40** | **73.13** | **88.04** | **40.92** | **48.03** | **85.06** | **44.93** | **66.53** | **80.03** |
| **16** | **57.37** | **74.24** | **91.62** | **44.48** | **53.15** | **86.28** | **27.46** | **47.98** | **68.48** | **54.35** | **73.18** | **88.00** | **41.12** | **48.81** | **84.82** | **45.43** | **65.30** | **79.94** |
| **32** | **57.96** | **75.93** | **92.01** | **43.12** | **53.57** | **86.80** | **30.25** | **49.62** | **70.68** | **53.62** | **72.41** | **87.63** | **07.18** | **26.28** | **75.18** | **06.12** | **29.91** | **63.56** |

Table 2: Automatic evaluation metrics for the OPUS-MT models and the SMaLL-100 MT model

the unique linguistic structures of the UR language. The reduced scores (Appendices B & C) indicated the Opus-mt-en-ur model experienced difficulties in capturing the complexity of the UR language, likely due to distinct syntactic characteristics.

**SMaLL-100** In contrast, the SMaLL-100 model demonstrated improved performance on the EN-UR language pair, surpassing the Opus-mt-en-ur at smaller batch sizes, suggesting better adaptability to the unique linguistic structures of UR. However, performance declined significantly at a batch size of 32, resulting in low scores. (Appendices B & C) The model exhibited performance trends similar to the OPUS-MT models across the EN-FI and EN-ZH language pairs. On the EN-FI pair, the model achieved competitive BLEU and CHRF scores, though slightly lower than the Opus-mt-en-fi model (Table 2). The SMaLL-100 model demonstrated comparable performance to Opus-mt-en-zh on smaller batch sizes, with only a slight decline in BLEU and COMET scores. Translation quality declined, however, on the EN-ZH pair at a batch size of 32. (Appendices B & C)

Based on the results (Table 2, Appendices B & C), we hypothesize that the Opus-MT models outperformed the SMaLL-100 model due to language-specific training, which enabled optimization and improved translation quality.

## 6 Analysis

**Automatic Evaluation** Overall, the Opus-mt-en-fi model demonstrated robust performance on the EN-FI language pair (Appendices B & D). The Opus-mt-en-fi achieved the highest BLEU scores (58.25, 57.37, and 57.97) on the MT task, closely followed by the SMaLL-100 model. Both models maintained strong consistency on the EN-FI language pair (Table 2). Similarly, the Opus-mt-en-zh model demonstrated satisfactory translation quality across all batch sizes. The SMaLL-100 model matched the performance stability at smaller batch sizes (8 and 16). However, performance declined at batch size 32 on the EN-ZH corpus, which showed a reduction in effectiveness and translation quality (Appendix B). Notably, the SMaLL-100 model demonstrated stronger performance than the Opus-mt-en-ur model at smaller batch sizes (8 and 16), which suggested the multilingual model was more effective in capturing the unique language patterns of the UR language. Performance declined significantly at batch size 32, mirroring patterns observed in the EN-ZH language pair. (Appendix B)

| Language Pair | Correct (%) | Partially Correct (%) | Incorrect (%) |
|---|---|---|---|
| **In-domain** | | | |
| EN - FI | 67.34 | 25.17 | 07.50 |
| EN - ZH | 45.85 | 06.83 | 47.32 |
| EN - UR | 26.57 | 60.78 | 12.65 |
| **Out-of-domain** | | | |
| EN - FI | 54.98 | 09.62 | 35.40 |
| EN - ZH | 20.27 | 06.19 | 73.54 |
| EN - UR | 06.19 | 21.99 | 71.82 |

Table 3: Percentage-based human evaluation across language pairs

**Human Evaluation** A qualitative analysis guided the human evaluation to determine whether the translations were correct, partially correct, or incorrect (Table 3). The evaluators observed multiple gold-standard translations (Appendices D, E & F) and some discrepancies (Appendix G) across the EN-FI, EN-ZH, and EN-UR language pairs, highlighting differences in generalization among the models. In a few cases, the human evaluators noticed that the models generated synonyms for some cancer terminologies, skipped translations, and produced grammatical errors (Appendix G).

**Skipped Translations** In some instances, no translation occurred across the language pairs, indicating limitations in the capacity of the Opus-MT and SMaLL-100 models to convert source references into the target language due to the unique morphological structure of each language.

For instance, the term *Topotecan* remained in its EN form, not matching the ZH reference. (Appendix G)

**Grammatical Errors** Punctuation and spacing errors occurred during the translation of some terminologies. While the models translated the cancer terminologies accurately, the generated output did not include the unique grammatical rule of the specific target language. (Appendix G)

**Ambiguous Terms** Some translations featured incorrect word order or introduced extraneous tokens, which distorted the intended meaning of the target reference. Additionally, an extra token generated during translation distorted the ZH reference for the term *Vancomycin-resistant Enterococcus*. Similarly, in the UR language pair, the term *advanced age* did not align with the target reference, reflecting a syntactic and semantic mismatch of the target language. (Appendix G)

## 7 Limitations

A significant limitation of the task was the size of the annotated corpus. The C A N C E R corpus included limited data in only three languages out of more than 7,000 spoken worldwide, which restricted the scope of the findings and applicability to broader multilingual contexts. While model performance was satisfactory overall, the data constraint likely contributed to the instability observed in the multilingual SMaLL-100 model at higher batch sizes, where translation quality degraded. Additionally, the UR language presented unique challenges due to its right-to-left script, which may have complicated the tokenization process. The limitations necessitate the need to expand the corpus and further experiment with optimizing techniques and models to improve translation quality across languages.

## 8 Conclusion and Future Work

In this paper, we took the first step towards advancing multilingual cancer patient education. The C A N C E R corpus serves as a benchmark resource for evaluating the translation of cancer terminology across languages. The findings inform efforts to improve multilingual cancer patient education, supporting non-native English speakers in understanding critical health information. We demonstrated that retraining on limited high-quality parallel data (Shin et al., 2020) can improve translation quality (Table 2). In future work, we aim to expand the C A N C E R corpus by incorporating a broader spectrum of low and high-resource languages and exploring varying techniques and NMT models to optimize performance, mainly in underrepresented languages.


## Acknowledgements

We want to express our gratitude to Deputy Head of School (Research) Dr. Camille Dickson-Deane and Senior Lecturer Dr. Amara Atif of the University of Technology Sydney for the final review of the paper and Associate Professors Rob van der Goot and Christian Hardmeier of the IT University of Copenhagen for feedback on the initial version. We are also grateful to the medical practitioners who reviewed segments of the annotated data to verify the target references of some terminologies.


## References


Md Mahfuz Ibn Alam, Ivana Kvapilíková, Antonios Anastasopoulos, Laurent Besacier, Georgiana Dinu, Marcello Federico, Matthias Gallé, Kweonwoo Jung, Philipp Koehn, and Vassilina Nikoulina. 2021. Findings of the wmt shared task on machine translation using terminologies. In *Proceedings of the Sixth Conference on Machine Translation*, pages 652–663.

Shuang Ao and Xeno Acharya. 2021. Learning ulmfit and self-distillation with calibration for medical dialogue system. *arXiv preprint arXiv:2107.09625*.

Freddie Bray, Mathieu Laversanne, Elisabete Weiderpass, and Isabelle Soerjomataram. 2021. The ever-increasing importance of cancer as a leading cause of premature death worldwide. *Cancer*, 127(16):3029–3030.

Pengshan Cai, Zonghai Yao, Fei Liu, Dakuo Wang, Meghan Reilly, Huixue Zhou, Lingxi Li, Yi Cao, Alok Kapoor, Adarsha Bajracharya, Dan Berlowitz, and Hong Yu. 2023. PaniniQA: Enhancing patient education through interactive question answering. *Transactions of the Association for Computational Linguistics*, 11:1518–1536.

Andre Castilla, Alice Bacic, and Sergio Furuie. 2005. Machine translation on the medical domain: the role of bleu/nist and meteor in a controlled vocabulary setting. In *Proceedings of Machine Translation Summit X: Papers*, pages 47–54.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Ona de Gibert Bonet, Ksenia Kharitonova, Blanca Calvo Figueras, Jordi Armengol-Estapé, and Maite Melero. 2022. Quality versus quantity: Building catalan-english mt resources. In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 59–69.

Marie Escribe. 2019. Human evaluation of neural machine translation: The case of deep learning. In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019)*, pages 36–46.

Muhammad Jauharul Fuady, Alvalen Shafelbilyunazra, Agusta Rakhmat Taufani, Yogi Dwi Mahandi, and Shofiyah Al Idrus. 2024. Hyperparameter optimization for transformer-based translation models on low-power devices. In *2024 Beyond Technology Summit on Informatics International Conference (BTS-I2C)*, pages 631–636. IEEE.

James E Harrison, Stefanie Weber, Robert Jakob, and Christopher G Chute. 2021. Icd-11: an international classification of diseases for the twenty-first century. *BMC medical informatics and decision making*, 21:1–10.

Paula Sofia Herrera-Espejel and Stefan Rach. 2023. The use of machine translation for outreach and health communication in epidemiology and public health: scoping review. *JMIR Public Health and Surveillance*, 9(1):e50814.

Ramunė Kasperė, Jurgita Mikelionienė, and Dalia Venckienė. 2023. Medical terminology issues: a feasibility study of machine translation in a low-resource language. *SKASE Journal of Translation and Interpretation*, 16(2):5–22.

Ramunė Kasperė, Jurgita Mikelionienė, and Dalia Venckienė. 2024. Medical terminology issues: a feasibility study of machine translation in a low-resource language. *SKASE Journal of Translation and Interpretation*, 1:5–22.

Kate R Lorig and Halsted R Holman. 2003. Self-management education: history, definition, outcomes, and mechanisms. *Annals of behavioral medicine*, 26(1):1–7.

Christian Lovis, Robert Baud, Anne-Marie Rassinoux, Pierre-André Michel, and Jean-Raoul Scherrer. 1998. Medical dictionaries for patient encoding systems: a methodology. *Artificial intelligence in medicine*, 14(1-2):201–214.

Hetong Ma, Feihong Yang, Jiansong Ren, Ni Li, Min Dai, Xuwen Wang, An Fang, Jiao Li, Qing Qian, and Jie He. 2020. Eccparacorp: a cross-lingual parallel corpus towards cancer education, dissemination and application. *BMC Medical Informatics and Decision Making*, 20:1–12.

Peter MacCallum. 2024. Everyday cancer words and terms: A to z. Victoria, Australia. Available at https://www.petermac.org/patients-and-carers/information-and-resources/a-z-of-everyday-cancer-words-and-terms.

Ruth McCorkle, Elizabeth Ercolano, Mark Lazenby, Dena Schulman-Green, Lynne S Schilling, Kate Lorig, and Edward H Wagner. 2011. Self-management: Enabling and empowering patients living with cancer as a chronic illness. *CA: a cancer journal for clinicians*, 61(1):50–62.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. SMaLL-100: Introducing shallow multilingual machine translation model for low-resource languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8348–8359, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Prashanth Nayak, Rejwanul Haque, and Andy Way. 2020. The adapt' s submissions to the wmt20 biomedical translation task.

Prashanth Nayak, John Kelleher, Rejwanul Haque, and Andy Way. 2023. Instance-based domain adaptation for improving terminology translation. In *Proceedings of Machine Translation Summit XIX, Vol. 1: Research Track*, pages 222–234.

David Oniani, Sreekanth Sreekumar, Renuk DeAlmeida, Dinuk DeAlmeida, Vivian Hui, Young Ji Lee, Yiye Zhang, Leming Zhou, and Yanshan Wang. 2023. Toward improving health literacy in patient education materials with neural machine translation models. *AMIA Summits on Translational Science Proceedings*, 2023:418.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Chanjun Park, Sugyeong Eo, Hyeonseok Moon, and Heui-Seok Lim. 2021. Should we find another model?: Improving neural machine translation performance with one-piece tokenization method without model modification. In *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: human language technologies: Industry papers*, pages 97–104.

Maja Popović. 2015. chrf: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference* *on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Hoo-Chang Shin, Yang Zhang, Evelina Bakhturina, Raul Puri, Mostofa Patwary, Mohammad Shoeybi, and Raghav Mani. 2020. Biomegatron: larger biomedical domain language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4700–4706.

Konstantinos Skianis, Yann Briand, and Florent Desgrippes. 2020. Evaluation of machine translation methods applied to medical terminologies. In *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, pages 59–69, Online. Association for Computational Linguistics.

Shivam R Solanki and Drupad K Khublani. 2024. Generative large language models. In *Generative Artificial Intelligence: Exploring the Power and Potential of Generative AI*, pages 229–296. Springer.

Jörg Tiedemann and Ona de Gibert. 2023. The OPUS-MT dashboard – a toolkit for a systematic evaluation of open machine translation models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 315–327, Toronto, Canada. Association for Computational Linguistics.

Mohamed Ugas, Meredith Giuliani, and Janet Papadakos. 2024. When is good, good enough? on considerations of machine translation in patient education. *Journal of Cancer Education*, 39(5):474–476.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. Document-level machine translation with large language models. *arXiv preprint arXiv:2304.02210*.

Xuan Zhang, Navid Rajabi, Kevin Duh, and Philipp Koehn. 2023. Machine translation with large language models: Prompting, few-shot learning, and fine-tuning with qlora. In *Proceedings of the Eighth Conference on Machine Translation*, pages 468–481.

# Appendix

## A Hyperparameters

| Model | dropout rate | learning rate grid | weight decay | batch size | epochs | optimizer |
|---|---|---|---|---|---|---|
| **Opus-MT** | 0.1 | 1e-05, 3e-05, 5e-05, 7e-05, 1e-04, 3e-04, 4e-04, 5e-04, 2e-04, **6e-04**, 7e-04 | 0.01 | **8**, 16, 32 | 3 | adamw |
| **SMaLL-100** | 0.1 | 1e-05, 3e-05, 5e-05, **7e-05**, 1e-04, 3e-04, 4e-04, 5e-04, 2e-04, 6e-04, 7e-04 | 0.01 | **8**, 16, 32 | 3 | adamw |

Table 4: Fine-tuning hyperparameters, best in bold

## B Model Performance

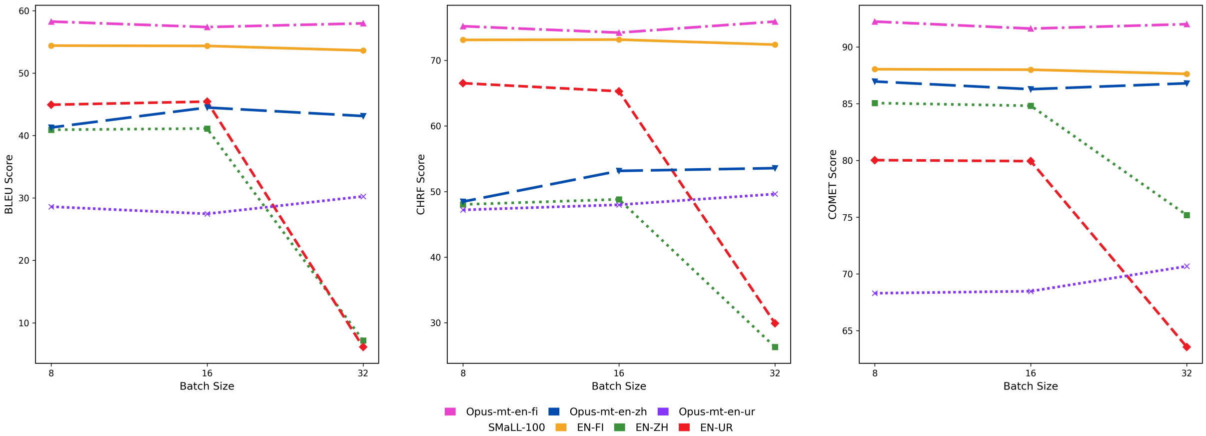



Figure 2: Evaluation metrics of the Opus-MT models and the SMaLL-100 model

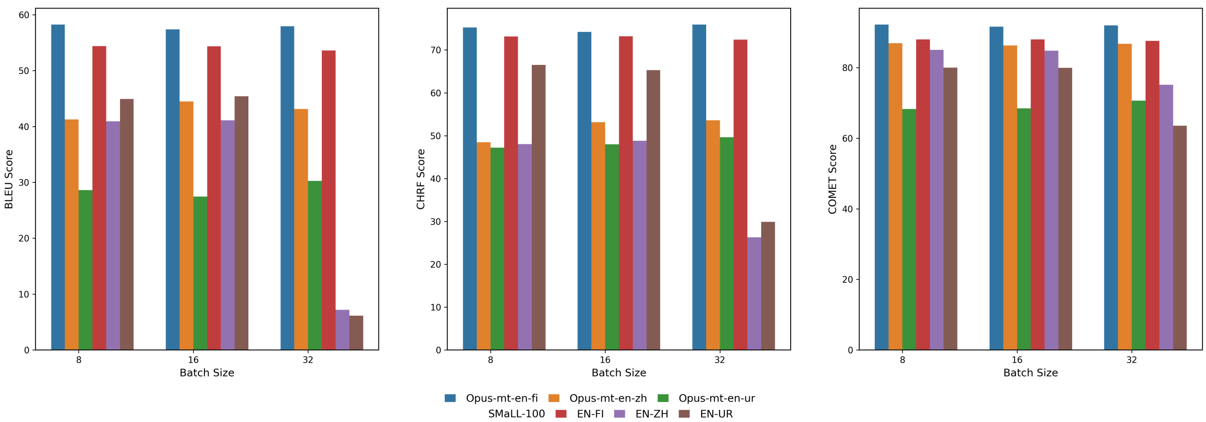



Figure 3: Comparison performance of the Opus-MT models and the SMaLL-100 model

## C Automatic Evaluation

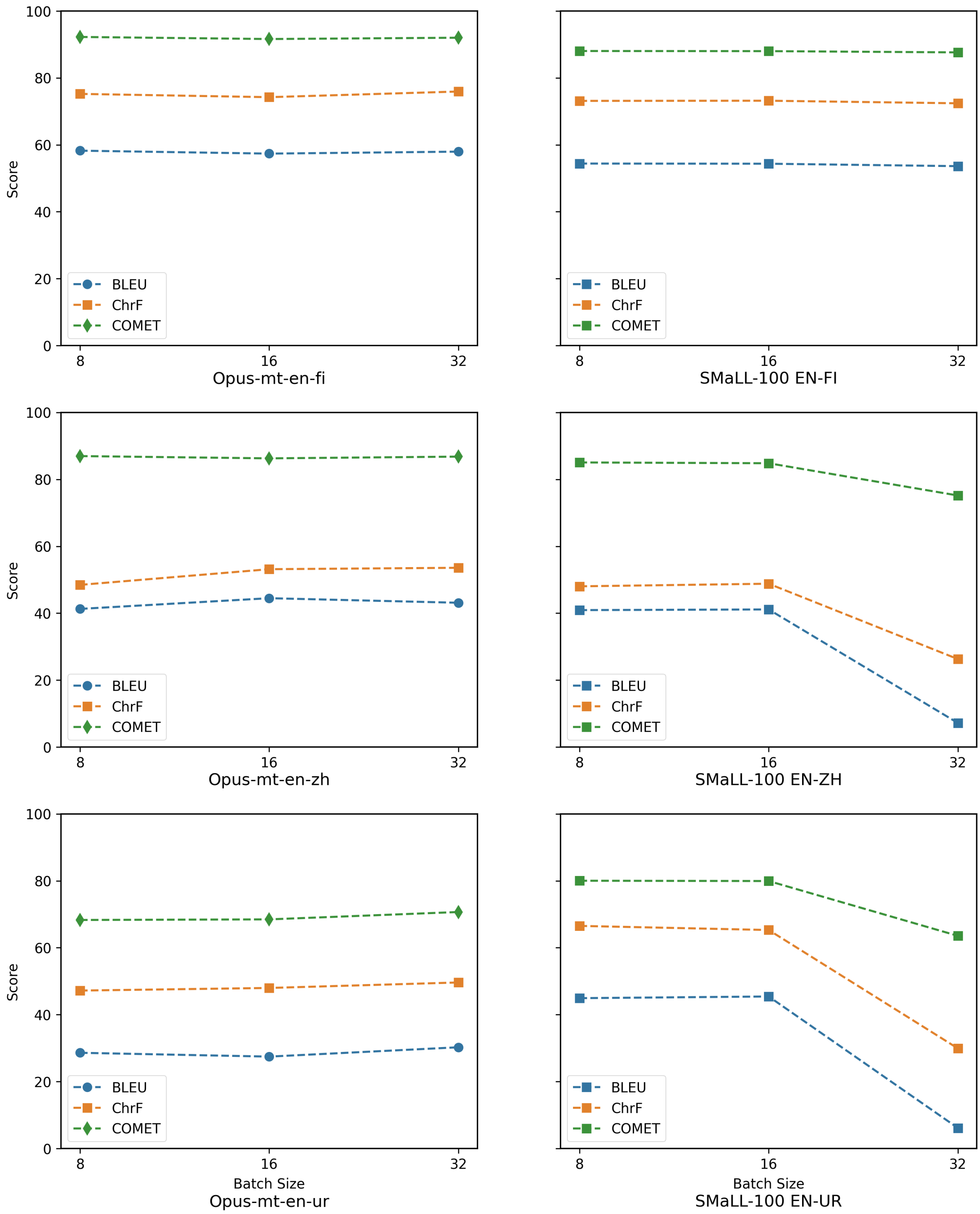



Figure 4: Performance metrics across language pairs

# D Gold-Standard EN–FI Translations

| Source (EN) | Reference (FI) | Target (FI) |
|---|---|---|
| Abdominal pain | vatsakipu | vatsakipu |
| Adverse effect | haitallinen vaikutus | haitallinen vaikutus |
| AFP levels | AFP-tasot | AFP-tasot |
| Alternative therapy | vaihtoehtohoito | vaihtoehtohoito |
| Anastrozole | anastrotsoli | anastrotsoli |
| Anatomy | anatomia | anatomia |
| Barium enema | bariumperäruiske | bariumperäruiske |
| Beckwith-Wiedemann | Beckwith-Wiedemann | Beckwith-Wiedemann |
| Blood count | verimäärä | verimäärä |
| Breast reconstruction | rinnan korjaus | rinnan korjaus |
| Bronchioloalveolar carcinoma | pienisoluinen keuhkosyöpä | pienisoluinen keuhkosyöpä |
| Burkitt lymphoma | Burkittin imukudossyöpä | Burkittin imukudossyöpä |
| Cancer staging | syövän vaiheistus | syövän vaiheistus |
| Chemotherapy | kemoterapia | kemoterapia |
| CT Colonography | paksusuolen CT-kuvantaminen | paksusuolen CT-kuvantaminen |
| CT imaging | CT-kuvantaminen | CT-kuvantaminen |
| diagnostic imaging | diagnostinen kuvantaminen | diagnostinen kuvantaminen |
| Digestive system | ruoansulatusjärjestelmä | ruoansulatusjärjestelmä |
| Epithelioid Hemangioendothelioma | solukudoskasvain | solukudoskasvain |
| Estrogen-only therapy | vain estrogeenihoito | vain estrogeenihoito |
| Febrile neutropenia | kuumeinen neutropenia | kuumeinen neutropenia |
| Follow-up | seuranta | seuranta |
| General anaesthetic | yleispuudutus | yleispuudutus |
| Germ cells | sukusolut | sukusolut |
| Hodgkin’s lymphoma | Hodgkin-lymfooma | Hodgkin-lymfooma |
| High-grade dysplasia | korkea-asteinen epänormaali solukasvu | korkea-asteinen epänormaali solukasvu |
| Inflammatory carcinoma | tulehdusperäinen syöpä | tulehdusperäinen syöpä |
| Intestinal | suolisto | suolisto |
| key hole surgery | avainaukkoleikkaus | avainaukkoleikkaus |
| Kaposi sarcoma | Kaposi-sarkooma | Kaposi-sarkooma |
| Laparoscopic surgery | laparoskooppinen leikkaus | laparoskooppinen leikkaus |
| Lymph glands | imusolmukkeet | imusolmukkeet |
| Magnetic resonance imaging | magneettikuvaus | magneettikuvaus |
| Medical oncology | lääketieteellinen onkologia | lääketieteellinen onkologia |
| Neoadjuvant treatment | uusi hoidon tehokkuutta parantava hoito | uusi hoidon tehokkuutta parantava hoito |
| Nuclear medicine | isotooppilääke | isotooppilääke |
| Occult carcinoma | selittämätön syöpä | selittämätön syöpä |
| Oxaliplatin | oksaliplatiini | oksaliplatiini |
| Primary lymphoma | ensisijainen imukudossyöpä | ensisijainen imukudossyöpä |
| Palliative therapy | palliatiivinen hoito | palliatiivinen hoito |
| Radioactive tracer | radioaktiivinen merkkiaine | radioaktiivinen merkkiaine |
| Recurrent cancer | uusiutuva syöpä | uusiutuva syöpä |
| Sepsis pathway | sepelvaltimointerventioreitti | sepelvaltimointerventioreitti |
| Stage 4 | vaihe 4 | vaihe 4 |
| Tissue biopsy | kudoskoepalan otto | kudoskoepalan otto |
| Tumor location | kasvaimen sijainti | kasvaimen sijainti |
| Unknown | tuntematon | tuntematon |
| use of statins | statiinien käyttö | statiinien käyttö |
| Vascular invasion | verisuonen invaasio | verisuonen invaasio |
| Variants | muunnokse | muunnokset |
| Weakness | heikkous | heikkous |
| Weight gain | painonnousu | painonnousu |
| X-ray | röntgenkuvaus | röntgenkuvaus |

Table 5: A subset of accurately translated EN–FI cancer terminologies assessed with human evaluation

## E Gold-Standard EN–ZH Translations

| Source (EN) | Reference (ZH) | Target (ZH) |
|---|---|---|
| Adenopathy | 腺病 | 腺病 |
| Anaemia | 贫血 | 贫血 |
| Antibody | 抗体 | 抗体 |
| Anus | 肛门 | 肛门 |
| Artery | 动脉 | 动脉 |
| Assess | 评估 | 评估 |
| Atrophy | 萎缩 | 萎缩 |
| Benign | 良性 | 良性 |
| Cells | 细胞 | 细胞 |
| Colon | 结肠 | 结肠 |
| Dialysis | 透析 | 透析 |
| Diarrhoea | 腹泻 | 腹泻 |
| Embolism | 栓塞 | 栓塞 |
| Excision | 切除术 | 切除术 |
| Faeces | 粪便 | 粪便 |
| Gynaecology | 妇科 | 妇科 |
| Hypertension | 高血压 | 高血压 |
| Hysterectomy | 子宫切除术 | 子宫切除术 |
| Incontinence | 失禁 | 失禁 |
| Isotope | 同位素 | 同位素 |
| Laparoscopy | 腹腔镜 | 腹腔镜 |
| Lymph | 淋巴结 | 淋巴 |
| Lymphoedema | 淋巴水肿 | 淋巴水肿 |
| Lymphoma | 淋巴瘤 | 淋巴瘤 |
| Mastectomy | 乳房切除术 | 乳房切除术 |
| Metastasis | 转移 | 转移 |
| Oedema | 水肿 | 水肿 |
| Oncology | 肿瘤学 | 肿瘤学 |
| Pathology | 病理学 | 病理学 |
| Rectum | 直肠 | 直肠 |
| Recurrence | 复发 | 复发 |
| Relapse | 复发 | 复发 |
| Risk | 风险 | 风险 |
| Sarcoma | 肉瘤 | 肉瘤 |
| Screening | 筛查 | 筛查 |
| Side-effect | 副作用 | 副作用 |
| Specimen | 样本 | 标本 |
| Staging | 分期 | 分期 |
| Surgery | 手术 | 手术 |
| Tissue | 组织 | 组织 |
| Tumour | 肿瘤 | 肿瘤 |
| Urethra | 尿道 | 尿道 |
| adjuvant chemotherapy | 辅助化疗 | 辅助化疗 |
| allergic reaction | 过敏反应 | 过敏反应 |
| carcinoma in situ | 原位癌 | 原位癌 |
| chronic pain | 慢性疼痛 | 慢性疼痛 |
| clinical trial | 临床试验 | 临床试验 |
| digestive system | 消化系统 | 消化系统 |
| germ cells | 生殖细胞 | 生殖细胞 |
| informed consent | 知情同意 | 知情同意 |
| local anaesthetic | 局部麻醉 | 局部麻醉 |
| neoadjuvant treatment | 新辅助治疗 | 新辅助治疗 |
| quality of life | 生活质量 | 生活质量 |
| sentinel node | 前哨淋巴结 | 前哨淋巴结 |
| small bowel | 小肠 | 小肠 |
| soft tissue | 软组织 | 软组织 |

Table 6: A subset of accurately translated EN–ZH cancer terminologies assessed with human evaluation

## F Gold-Standard EN–UR Translations

| Source (EN) | Reference (UR) | Target (UR) |
|---|---|---|
| Ablation Techniques | مٹانے کے طریقے | مٹانے کے طریقے |
| Acute hepatitis | سوزش کی جگر تیز | سوزش کی جگر تیز |
| alternative therapy | متبادل علاج | متبادل علاج |
| Anorexia | بھوک کی کمی | بھوک کی کمی |
| Better tolerability | بہتر برداشت (علاج) | بہتر برداشت (علاج) |
| Breast Self-examination | چھاتی کا خود معائنہ | چھاتی کا خود معائنہ |
| cancer prevention | سرطان کی روک تھام | سرطان کی روک تھام |
| Chemotherapy risks | کیمیائی علاج کے خطرات | کیمیائی علاج کے خطرات |
| chronic pain | دائمی درد | دائمی درد |
| Clinical trials | طبّی تجربات | طبّی تجربات |
| Combination chemotherapy | مجموعی کیمیائی علاج | مجموعی کیمیائی علاج |
| Contamination: None | آلودگی : کوئی نہیں | آلودگی : کوئی نہیں |
| Contralateral Disease | مخالف طرفی بیماری | مخالف طرفی بیماری |
| Diagnostic imaging | تشخیصی امیجنگ | تشخیصی امیجنگ |
| Discomfort | تکلیف | تکلیف |
| Dominant Geographical Areas | غالب جغرافیائی علاقے | غالب جغرافیائی علاقے |
| Dose/Trial Drug | آزمائشی/خوراک دوا | آزمائشی/خوراک دوا |
| Early pregnancy | ابتدائی حمل | ابتدائی حمل |
| Environmental factors | ماحولیاتی عوامل | ماحولیاتی عوامل |
| Excessive alcohol use | ضرورت سے زیادہ شراب کا استعمال | ضرورت سے زیادہ شراب کا استعمال |
| Family history | خاندانی تاریخ | خاندانی تاریخ |
| Follow-up | تجزیہ: پیروی | تجزیہ: پیروی |
| General Information About Small Cell Lung Cancer | چھوٹے سیل پھیپھڑوں کے سرطان کے بارے میں عمومی معلومات | چھوٹے سیل پھیپھڑوں کے سرطان کے بارے میں عمومی معلومات |
| Genetic risk factors | جینیاتی خطرے کے عوامل | جینیاتی خطرے کے عوامل |
| Hepatitis B | جگر کی سوزش بی | جگر کی سوزش بی |
| Hoarseness | آواز کا بیٹھ جانا | آواز کا بیٹھ جانا |
| Incidence and Mortality | واقعات اور اموات | واقعات اور اموات |
| Internal Validity : Fair | داخلی توثیق: معتدل | داخلی توثیق: معتدل |
| International Comparisons | بین الاقوامی موازنہ | بین الاقوامی موازنہ |
| Local radiation therapy | مقامی ریڈی ایشن علاج | مقامی ریڈی ایشن علاج |
| Low-birth-weight infants | کم پیدائشی وزن کے نوزائیدہ | کم پیدائشی وزن کے نوزائیدہ |
| Male breast cancer is rare | مردوں کے چھاتی کا سرطان نایاب ہے | مردوں کے چھاتی کا سرطان نایاب ہے |
| Occult NSCLC | پوشیدہ این ایس سی ایل سی کا علاج | پوشیدہ این ایس سی ایل سی کا علاج |
| Other risk factors | دیگر خطرے کے عوامل | دیگر خطرے کے عوامل |
| Overdiagnosis | ضرورت سے زیادہ تشخیص | ضرورت سے زیادہ تشخیص |
| Palliative therapy | تسکینی علاج | تسکینی علاج |
| Pathologic Classification | مرضیاتی درجہ بندی | مرضیاتی درجہ بندی |
| Patient Evaluation | مریض کا جائزہ | مریض کا جائزہ |
| Physical activity | جسمانی سرگرمی | جسمانی سرگرمی |
| Population-level interventions | آبادی کی سطح پر مداخلت | آبادی کی سطح پر مداخلت |
| Presurgical chemotherapy | سرجری سے پہلے کی کیمیائی علاج | سرجری سے پہلے کی کیمیائی علاج |
| Prognosis–legacy | پیش گوئی (پرانا) | پیش گوئی (پرانا) |
| recurrent rectal cancer | بار بار ہونے والا مستقیمی سرطان | بار بار ہونے والا مستقیمی سرطان |
| Screening Intervention | اسکریننگ مداخلت | اسکریننگ مداخلت |
| Special Populations | مخصوص آبادی | مخصوص آبادی |
| Stage explanation–legacy | مرحلے کی وضاحت (پرانا) | مرحلے کی وضاحت (پرانا) |
| Standard treatment | معیاری علاج | معیاری علاج |
| Study Design: Evidence obtained from large databases | مطالعہ کا ڈیزائن: بڑے ڈیٹا بیس سے حاصل شواہد | مطالعہ کا ڈیزائن: بڑے ڈیٹا بیس سے حاصل شواہد |
| The comparison group was not actively followed | موازنہ گروپ کی فعال نگرانی نہیں کی گئی | موازنہ گروپ کی فعال نگرانی نہیں کی گئی |
| The overall 5-year survival rate is 64% | کل 5 سالہ بقا کی شرح 64% ہے | کل 5 سالہ بقا کی شرح 64% ہے |
| To assess the efficacy of initial therapy | ابتدائی علاج کی کارکردگی کا جائزہ لینا | ابتدائی علاج کی کارکردگی کا جائزہ لینا |
| Tumor Characteristics | رسولی کی خصوصیات | رسولی کی خصوصیات |
| Weight gain | وزن میں اضافہ | وزن میں اضافہ |
| Who is at Risk | خطرے میں کون ہے | خطرے میں کون ہے |

Table 7: A subset of accurately translated EN–UR cancer terminologies assessed with human evaluation

## G Translation Errors

| Error Type | Source | Reference | Target |
|---|---|---|---|
| **Skipped Translations** | Exemestane | antineoplastinen lääke | exemestane |
| | radiation therapist | 放射治疗师 | |
| | Deaths: 10,990 | اموات: 10,990 | اموات: 10,99 |
| **Grammatical Errors** | GP (general practitioner) | GP（yleislääkäri） | GP(yleinen lääkäri) |
| | Consistency: Consistent | 一致性：一致 | 一致性: 一致 |
| **Ambiguous Terms** | Ablation Techniques | ablaatiotekniikat | kudospoistotekniikat |
| | Vancomycin Resistant Enterococcus | 万古霉素耐药肠球菌 | the 霉素抗性肠杆菌 |
| | Advanced age | عمر رسیدگی | دور عمر |

Table 8: Some translation errors observed with human evaluation

# Author Index